

```
SSSSSSSSSSSSS  MMM      MMM      GGGGGGGGGGGG  RRRRRRRRRRRR  TTTTTTTTTTTTTT  LLL
SSSSSSSSSSSSS  MMM      MMM      GGGGGGGGGGGG  RRRRRRRRRRRR  TTTTTTTTTTTTTT  LLL
SSSSSSSSSSSSS  MMM      MMM      GGGGGGGGGGGG  RRRRRRRRRRRR  TTTTTTTTTTTTTT  LLL
SSS            MMMMMM  MMMMMM  GGG            RRR      RRR      TTT      LLL
SSS            MMMMMM  MMMMMM  GGG            RRR      RRR      TTT      LLL
SSS            MMMMMM  MMMMMM  GGG            RRR      RRR      TTT      LLL
SSS            MMM      MMM      GGG            RRR      RRR      TTT      LLL
SSS            MMM      MMM      GGG            RRR      RRR      TTT      LLL
SSS            MMM      MMM      GGG            RRR      RRR      TTT      LLL
SSS            MMM      MMM      GGG            RRR      RRR      TTT      LLL
SSSSSSSSSSS    MMM      MMM      GGG            RRRRRRRRRRRR  TTT      LLL
SSSSSSSSSSS    MMM      MMM      GGG            RRRRRRRRRRRR  TTT      LLL
SSSSSSSSSSS    MMM      MMM      GGG            RRRRRRRRRRRR  TTT      LLL
SSS            MMM      MMM      GGG      GGGGGGGGGG  RRR      RRR      TTT      LLL
SSS            MMM      MMM      GGG      GGGGGGGGGG  RRR      RRR      TTT      LLL
SSS            MMM      MMM      GGG      GGGGGGGGGG  RRR      RRR      TTT      LLL
SSS            MMM      MMM      GGG      GGG      GGG  RRR      RRR      TTT      LLL
SSS            MMM      MMM      GGG      GGG      GGG  RRR      RRR      TTT      LLL
SSS            MMM      MMM      GGG      GGG      GGG  RRR      RRR      TTT      LLL
SSS            MMM      MMM      GGG      GGG      GGG  RRR      RRR      TTT      LLL
SSS            MMM      MMM      GGG      GGG      GGG  RRR      RRR      TTT      LLL
SSSSSSSSSSSSS  MMM      MMM      GGGGGGGGGG  RRR      RRR      TTT      LLLLLLLLLLLLLLLL
SSSSSSSSSSSSS  MMM      MMM      GGGGGGGGGG  RRR      RRR      TTT      LLLLLLLLLLLLLLLL
SSSSSSSSSSSSS  MMM      MMM      GGGGGGGGGG  RRR      RRR      TTT      LLLLLLLLLLLLLLLL
```

[illegible]

```
1 0001 0 MODULE SMG$STRING TABLES( XTITLE 'TPARSE tables for string capabilities'
2 0002 0 IDENT = '1-003' ! File: SMGSTRTAB.B32 Edit: PLL1003
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
10 0010 1 * ALL RIGHTS RESERVED.
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
17 0017 1 * TRANSFERRED.
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
21 0021 1 * CORPORATION.
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1
30 0030 1 ++
31 0031 1 FACILITY: Screen Management
32 0032 1
33 0033 1 ABSTRACT:
34 0034 1
35 0035 1 This module contains the LIB$TPARSE state tables used to parse
36 0036 1 string capabilities in an ascii TERMTABLE.TXT file.
37 0037 1
38 0038 1 ENVIRONMENT: User mode - AST reentrant
39 0039 1
40 0040 1 AUTHOR: P. Levesque CREATION DATE: 30-Jan-1984
41 0041 1
42 0042 1 MODIFIED BY:
43 0043 1
44 0044 1 1-001 - Original. PLL 30-Jan-1984
45 0045 1 -002 - Add SMG$K_BEGIN_NORMAL_RENDITION. PLL 15-Mar-1984
46 0046 1 Enhance state tables to accept only !UL FAO directive.
47 0047 1 Allow comments on lines not terminated by a comma.
48 0048 1 -003 - Change implementation of control characters. PLL 5-Jul-1984
49 0049 1 --
50 0050 1
```



```

52 0051 1 XSBTTL 'Declarations'
53 0052 1
54 0053 1 SWITCHES:
55 0054 1
56 0055 1
57 0056 1
58 0057 1 LINKAGES:
59 0058 1
60 0059 1 NONE
61 0060 1
62 0061 1 TABLE OF CONTENTS:
63 0062 1
64 0063 1
65 0064 1 FORWARD ROUTINE
66 0065 1 CONVERT_CONTROL,      ! convert ^ to control character
67 0066 1 CONVERT_ESCAPE,   ! convert $ to escape character
68 0067 1 EXPRESSION_ERROR, ! signal an invalid expression error
69 0068 1 END_ARITH_CAP,     ! end an arithmetic expression
70 0069 1 INIT_ARITH_CAP,    ! prepare for arithmetic expression
71 0070 1 INSERT_CARROT,     ! insert ^ into a string
72 0071 1 INSERT_DOLLAR,     ! insert $ into a string
73 0072 1 INSERT_EXCLAMATION, ! insert ! into a string
74 0073 1 INSERT_PARENTHESIS, ! insert ( into a string
75 0074 1 INVALID_DIRECTIVE, ! signal invalid directive
76 0075 1 NOT_STRING,        ! signal an unknown capability name
77 0076 1 STORE_OPERATOR,    ! store number & operator
78 0077 1 STORE_SUBSTITUTION; ! store % flag in arithmetic cap
79 0078 1
80 0079 1
81 0080 1 INCLUDE FILES:
82 0081 1
83 0082 1
84 0083 1 REQUIRE 'RTLIN:SMGPROLOG';      ! Defines psects, macros, etc.
85 0161 1
86 0162 1 LIBRARY 'RTLML:SMGTPALIB';      ! Definitions and macros used
87 0163 1                                ! to create TERMTABLE.EXE
88 0164 1 LIBRARY 'RTLTPAMAC';            ! TPARSE library of macros
89 0165 1
90 0166 1
91 0167 1 EQUATED SYMBOLS:
92 0168 1
93 0169 1
94 0170 1 LITERAL
95 0171 1 SINGLE_QUOTE = 'X'27';
96 0172 1 DOUBLE_QUOTE = 'X'22';
97 0173 1 LEFT_PAREN = 'X'28';
98 0174 1 RIGHT_PAREN = 'X'29';
99 0175 1
100 0176 1
101 0177 1 FIELDS:
102 0178 1
103 0179 1 NONE
104 0180 1
105 0181 1 PSECTS:
106 0182 1
107 0183 1
108 0184 1

```

SMG\$STRING_TABL TPARSE tables for string capabilities
1-003 Declarations

N 8
16-Sep-1984 01:22:35 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 13:10:04 [SMGRTL.SRC]SMGSTRTAB.B32;1

Page 3
(2)

```

: 109      0185 1 : EXTERNAL REFERENCES:
: 110      0186 1 :
: 111      0187 1 EXTERNAL ROUTINE
: 112      0188 1      OTSS$CVT_T1_L,      : convert ascii digits to integer
: 113      0189 1      SMG$$BLANKS_OFF,    : turn off flag to process blanks
: 114      0190 1      SMG$$BLANKS_ON,      : turn on flag to process blanks
: 115      0191 1      SMG$$COPY_CAP,       : copy a string capability into TERMTABLE
: 116      0192 1      SMG$$FLUSH_ARITHMETIC, : flush expression data to TERMTABLE.EXE
: 117      0193 1      SMG$$FLUSH_SAVED_BUFFER, : flush saved token to TERMTABLE.EXE
: 118      0194 1      SMG$$MISSING_END,     : signal error
: 119      0195 1      SMG$$NEXT_RECORD,     : get next record from TERMTABLE.TXT
: 120      0196 1      SMG$$SAVE_TOKEN_STRING, : store count & ptr to token
: 121      0197 1      SMG$$STORE_CAP_MASK,  : remember capability number
: 122      0198 1      SMG$$SYNTAX_ERROR;    : signal syntax error
: 123      0199 1
: 124      0200 1 EXTERNAL
: 125      0201 1      SMG$_ERRAT_LIN,      : error in line n at or near 'x'
: 126      0202 1      SMG$_INVDIR,        : invalid directive
: 127      0203 1      SMG$_INVEXP,        : invalid expression
: 128      0204 1      SMG$_MISTERNAM,     : missing terminal name
: 129      0205 1      SMG$_NOTSTRCAP,     : not a string capability
: 130      0206 1      SMG$_SYNERR;        : syntax error
: 131      0207 1
: 132      0208 1 EXTERNAL
: 133      0209 1      SMG$$MASK_ADR,       : used by TPARSE action routines
: 134      0210 1      SMG$$NEXT_NEGATIVE_NUMBER, : used to set terminal number
: 135      0211 1      SMG$$CURRENT_LINE,   : current input line - maintained
: 136      0212 1                          : for error messages
: 137      0213 1      SMG$$DATA_OFFSET;    : offset into data buffer
: 138      0214 1
: 139      0215 1 :
: 140      0216 1 : OWN STORAGE:
: 141      0217 1 :
: 142      0218 1 : NONE

```

```

144 0219 1 %SBTTL 'SMG$$$STRING_TABLES - TPARSE tables for string capabilities'
145 0220 1 |++
146 0221 1 | FUNCTIONAL DESCRIPTION:
147 0222 1 |
148 0223 1 |     The following are the state tables used to parse string
149 0224 1 |     capabilities in a terminal definition.
150 0225 1 |
151 0226 1 | --
152 0227 1 |
153 0228 1 $INIT_STATE (SMG$$A_STRING_STATES, SMG$$A_STRING_KEYWDS);
154 0229 1 | set up state tables, key words
155 0230 1 |
156 0231 1 | +
157 0232 1 | Begin scanning loop. Look for the start of a capability.
158 0233 1 | Skip over blanks and comments.
159 0234 1 | -
160 0235 1 |
161 P 0236 1 $STATE (BEGIN_SCAN,
162 PP 0237 1 | ((END_OF_LINE), BEGIN_SCAN, SMG$$NEXT_RECORD),
163 PP 0238 1 | ('!', BEGIN_SCAN, SMG$$NEXT_RECORD),
164 PP 0239 1 | ((CAPABILITY?), BEGIN_SCAN, SMG$$BLANKS_OFF),
165 P 0240 1 | (TPAS_LAMBDA, TPAS_EXIT)
166 0241 1 | );
167 0242 1 |
168 0243 1 | +
169 0244 1 | This state indicates the end of a line. A comment also signals the
170 0245 1 | end of a line.
171 0246 1 | -
172 0247 1 |
173 P 0248 1 $STATE (END_OF_LINE,
174 P 0249 1 | (TPAS_EOS, TPAS_EXIT),
175 P 0250 1 | ('!', TPAS_EXIT),
176 P 0251 1 | (TPAS_LAMBDA, TPAS_FAIL)
177 0252 1 | );
178 0253 1 |
179 0254 1 | +
180 0255 1 | Find the capability name and determine if it's one that we expect.
181 0256 1 | The string up to the '=' sign should be the capability name.
182 0257 1 | -
183 0258 1 |
184 P 0259 1 $STATE (CAPABILITY,
185 P 0260 1 | ((STRING_NAME), EQUALS_STRING, SMG$$BLANKS_OFF),
186 P 0261 1 | ('END', TPAS_FAIL),
187 P 0262 1 | ('BOOLEAN', TPAS_FAIL),
188 P 0263 1 | ('NUMERIC', TPAS_FAIL),
189 P 0264 1 | ('STRING', BEGIN_SCAN),
190 P 0265 1 | ('REQUIRE', TPAS_FAIL, SMG$$MISSING_END),
191 P 0266 1 | ('NAME', TPAS_FAIL, SMG$$MISSING_END),
192 P 0267 1 | (TPAS_SYMBOL, NOT_STRING)
193 0268 1 | );
194 0269 1 |
195 0270 1 | +
196 0271 1 | Determine if string capability name is valid.
197 0272 1 | -
198 0273 1 |
199 P 0274 1 $STATE (STRING_NAME,
200 P 0275 1 | ('BEGIN_ALTERNATE_CHAR', TPAS_EXIT, , SMG$K_BEGIN_ALTERNATE_CHAR, SMG$$MASK_ADR),

```



```

201 P 0276 1 ('BEGIN-BLINK', TPAS_EXIT, , SMG$K_BEGIN-BLINK, SMG$MASK_ADR),
202 P 0277 1 ('BEGIN-BOLD', TPAS_EXIT, , SMG$K_BEGIN-BOLD, SMG$MASK_ADR),
203 P 0278 1 ('BEGIN-DELETE-MODE', TPAS_EXIT, , SMG$K_BEGIN-DELETE-MODE, SMG$MASK_ADR),
204 P 0279 1 ('BEGIN-INSERT-MODE', TPAS_EXIT, , SMG$K_BEGIN-INSERT-MODE, SMG$MASK_ADR),
205 P 0280 1 ('BEGIN-LINE-DRAWING-CHAR', TPAS_EXIT, , SMG$K_BEGIN-LINE-DRAWING-CHAR, SMG$MASK_ADR),
206 P 0281 1 ('BEGIN-NORMAL-RENDITION', TPAS_EXIT, , SMG$K_BEGIN-NORMAL-RENDITION, SMG$MASK_ADR),
207 P 0282 1 ('BEGIN-REVERSE', TPAS_EXIT, , SMG$K_BEGIN-REVERSE, SMG$MASK_ADR),
208 P 0283 1 ('BEGIN-UNDERSCORE', TPAS_EXIT, , SMG$K_BEGIN-UNDERSCORE, SMG$MASK_ADR),
209 P 0284 1 ('BOTTOM-T-CHAR', TPAS_EXIT, , SMG$K_BOTTOM-T-CHAR, SMG$MASK_ADR),
210 P 0285 1 ('CLEAR-TAB', TPAS_EXIT, , SMG$K_CLEAR-TAB, SMG$MASK_ADR),
211 P 0286 1 ('CR-GRAPHIC', TPAS_EXIT, , SMG$K_CR-GRAPHIC, SMG$MASK_ADR),
212 P 0287 1 ('CROSS-CHAR', TPAS_EXIT, , SMG$K_CROSS-CHAR, SMG$MASK_ADR),
213 P 0288 1 ('CURSOR-DOWN', TPAS_EXIT, , SMG$K_CURSOR-DOWN, SMG$MASK_ADR),
214 P 0289 1 ('CURSOR-LEFT', TPAS_EXIT, , SMG$K_CURSOR-LEFT, SMG$MASK_ADR),
215 P 0290 1 ('CURSOR-NEXT-LINE', TPAS_EXIT, , SMG$K_CURSOR-NEXT-LINE, SMG$MASK_ADR),
216 P 0291 1 ('CURSOR-POSITION-REPORT', TPAS_EXIT, , SMG$K_CURSOR-POSITION-REPORT, SMG$MASK_ADR),
217 P 0292 1 ('CURSOR-PRECEEDING-LINE', TPAS_EXIT, , SMG$K_CURSOR-PRECEEDING-LINE, SMG$MASK_ADR),
218 P 0293 1 ('CURSOR-RIGHT', TPAS_EXIT, , SMG$K_CURSOR-RIGHT, SMG$MASK_ADR),
219 P 0294 1 ('CURSOR-UP', TPAS_EXIT, , SMG$K_CURSOR-UP, SMG$MASK_ADR),
220 P 0295 1 ('DARK-SCREEN', TPAS_EXIT, , SMG$K_DARK-SCREEN, SMG$MASK_ADR),
221 P 0296 1 ('DELETE-CHAR', TPAS_EXIT, , SMG$K_DELETE-CHAR, SMG$MASK_ADR),
222 P 0297 1 ('DELETE-LINE', TPAS_EXIT, , SMG$K_DELETE-LINE, SMG$MASK_ADR),
223 P 0298 1 ('DEVICE-ATTRIBUTES', TPAS_EXIT, , SMG$K_DEVICE-ATTRIBUTES, SMG$MASK_ADR),
224 P 0299 1 ('DOUBLE-HIGH-BOTTOM', TPAS_EXIT, , SMG$K_DOUBLE-HIGH-BOTTOM, SMG$MASK_ADR),
225 P 0300 1 ('DOUBLE-HIGH-TOP', TPAS_EXIT, , SMG$K_DOUBLE-HIGH-TOP, SMG$MASK_ADR),
226 P 0301 1 ('DOUBLE-WIDE', TPAS_EXIT, , SMG$K_DOUBLE-WIDE, SMG$MASK_ADR),
227 P 0302 1 ('DUPLICATE', TPAS_EXIT, , SMG$K_DUPLICATE, SMG$MASK_ADR),
228 P 0303 1 ('END-ALTERNATE-CHAR', TPAS_EXIT, , SMG$K_END-ALTERNATE-CHAR, SMG$MASK_ADR),
229 P 0304 1 ('END-BLINK', TPAS_EXIT, , SMG$K_END-BLINK, SMG$MASK_ADR),
230 P 0305 1 ('END-BOLD', TPAS_EXIT, , SMG$K_END-BOLD, SMG$MASK_ADR),
231 P 0306 1 ('END-DELETE-MODE', TPAS_EXIT, , SMG$K_END-DELETE-MODE, SMG$MASK_ADR),
232 P 0307 1 ('END-INSERT-MODE', TPAS_EXIT, , SMG$K_END-INSERT-MODE, SMG$MASK_ADR),
233 P 0308 1 ('END-LINE-DRAWING-CHAR', TPAS_EXIT, , SMG$K_END-LINE-DRAWING-CHAR, SMG$MASK_ADR),
234 P 0309 1 ('END-REVERSE', TPAS_EXIT, , SMG$K_END-REVERSE, SMG$MASK_ADR),
235 P 0310 1 ('END-UNDERSCORE', TPAS_EXIT, , SMG$K_END-UNDERSCORE, SMG$MASK_ADR),
236 P 0311 1 ('ERASE-TO-END-DISPLAY', TPAS_EXIT, , SMG$K_ERASE-TO-END-DISPLAY, SMG$MASK_ADR),
237 P 0312 1 ('ERASE-TO-END-LINE', TPAS_EXIT, , SMG$K_ERASE-TO-END-LINE, SMG$MASK_ADR),
238 P 0313 1 ('ERASE-WHOLE-DISPLAY', TPAS_EXIT, , SMG$K_ERASE-WHOLE-DISPLAY, SMG$MASK_ADR),
239 P 0314 1 ('ERASE-WHOLE-LINE', TPAS_EXIT, , SMG$K_ERASE-WHOLE-LINE, SMG$MASK_ADR),
240 P 0315 1 ('FF-GRAPHIC', TPAS_EXIT, , SMG$K_FF-GRAPHIC, SMG$MASK_ADR),
241 P 0316 1 ('HOME', TPAS_EXIT, , SMG$K_HOME, SMG$MASK_ADR),
242 P 0317 1 ('HORIZONTAL-BAR', TPAS_EXIT, , SMG$K_HORIZONTAL-BAR, SMG$MASK_ADR),
243 P 0318 1 ('HT-GRAPHIC', TPAS_EXIT, , SMG$K_HT-GRAPHIC, SMG$MASK_ADR),
244 P 0319 1 ('INDEX', TPAS_EXIT, , SMG$K_INDEX, SMG$MASK_ADR),
245 P 0320 1 ('INIT-STRING', TPAS_EXIT, , SMG$K_INIT-STRING, SMG$MASK_ADR),
246 P 0321 1 ('INSERT-CHAR', TPAS_EXIT, , SMG$K_INSERT-CHAR, SMG$MASK_ADR),
247 P 0322 1 ('INSERT-LINE', TPAS_EXIT, , SMG$K_INSERT-LINE, SMG$MASK_ADR),
248 P 0323 1 ('INSERT-PAD', TPAS_EXIT, , SMG$K_INSERT-PAD, SMG$MASK_ADR),
249 P 0324 1 ('KEY-0', TPAS_EXIT, , SMG$K_KEY-0, SMG$MASK_ADR),
250 P 0325 1 ('KEY-1', TPAS_EXIT, , SMG$K_KEY-1, SMG$MASK_ADR),
251 P 0326 1 ('KEY-2', TPAS_EXIT, , SMG$K_KEY-2, SMG$MASK_ADR),
252 P 0327 1 ('KEY-3', TPAS_EXIT, , SMG$K_KEY-3, SMG$MASK_ADR),
253 P 0328 1 ('KEY-4', TPAS_EXIT, , SMG$K_KEY-4, SMG$MASK_ADR),
254 P 0329 1 ('KEY-5', TPAS_EXIT, , SMG$K_KEY-5, SMG$MASK_ADR),
255 P 0330 1 ('KEY-6', TPAS_EXIT, , SMG$K_KEY-6, SMG$MASK_ADR),
256 P 0331 1 ('KEY-7', TPAS_EXIT, , SMG$K_KEY-7, SMG$MASK_ADR),
257 P 0332 1 ('KEY-8', TPAS_EXIT, , SMG$K_KEY-8, SMG$MASK_ADR),

```



```

258 P 0333 1 ('KEY_9', TPAS_EXIT, SMG$K_KEY_9, SMG$MASK_ADR),
259 P 0334 1 ('KEY_BACKSPACE', TPAS_EXIT, SMG$K_KEY_BACKSPACE, SMG$MASK_ADR),
260 P 0335 1 ('KEY_COMMA', TPAS_EXIT, SMG$K_KEY_COMMA, SMG$MASK_ADR),
261 P 0336 1 ('KEY_DOWN_ARROW', TPAS_EXIT, SMG$K_KEY_DOWN_ARROW, SMG$MASK_ADR),
262 P 0337 1 ('KEY_E1', TPAS_EXIT, SMG$K_KEY_E1, SMG$MASK_ADR),
263 P 0338 1 ('KEY_E2', TPAS_EXIT, SMG$K_KEY_E2, SMG$MASK_ADR),
264 P 0339 1 ('KEY_E3', TPAS_EXIT, SMG$K_KEY_E3, SMG$MASK_ADR),
265 P 0340 1 ('KEY_E4', TPAS_EXIT, SMG$K_KEY_E4, SMG$MASK_ADR),
266 P 0341 1 ('KEY_E5', TPAS_EXIT, SMG$K_KEY_E5, SMG$MASK_ADR),
267 P 0342 1 ('KEY_E6', TPAS_EXIT, SMG$K_KEY_E6, SMG$MASK_ADR),
268 P 0343 1 ('KEY_ENTER', TPAS_EXIT, SMG$K_KEY_ENTER, SMG$MASK_ADR),
269 P 0344 1 ('KEY_F1', TPAS_EXIT, SMG$K_KEY_F1, SMG$MASK_ADR),
270 P 0345 1 ('KEY_F2', TPAS_EXIT, SMG$K_KEY_F2, SMG$MASK_ADR),
271 P 0346 1 ('KEY_F3', TPAS_EXIT, SMG$K_KEY_F3, SMG$MASK_ADR),
272 P 0347 1 ('KEY_F4', TPAS_EXIT, SMG$K_KEY_F4, SMG$MASK_ADR),
273 P 0348 1 ('KEY_F5', TPAS_EXIT, SMG$K_KEY_F5, SMG$MASK_ADR),
274 P 0349 1 ('KEY_F6', TPAS_EXIT, SMG$K_KEY_F6, SMG$MASK_ADR),
275 P 0350 1 ('KEY_F7', TPAS_EXIT, SMG$K_KEY_F7, SMG$MASK_ADR),
276 P 0351 1 ('KEY_F8', TPAS_EXIT, SMG$K_KEY_F8, SMG$MASK_ADR),
277 P 0352 1 ('KEY_F9', TPAS_EXIT, SMG$K_KEY_F9, SMG$MASK_ADR),
278 P 0353 1 ('KEY_F10', TPAS_EXIT, SMG$K_KEY_F10, SMG$MASK_ADR),
279 P 0354 1 ('KEY_F11', TPAS_EXIT, SMG$K_KEY_F11, SMG$MASK_ADR),
280 P 0355 1 ('KEY_F12', TPAS_EXIT, SMG$K_KEY_F12, SMG$MASK_ADR),
281 P 0356 1 ('KEY_F13', TPAS_EXIT, SMG$K_KEY_F13, SMG$MASK_ADR),
282 P 0357 1 ('KEY_F14', TPAS_EXIT, SMG$K_KEY_F14, SMG$MASK_ADR),
283 P 0358 1 ('KEY_F15', TPAS_EXIT, SMG$K_KEY_F15, SMG$MASK_ADR),
284 P 0359 1 ('KEY_F16', TPAS_EXIT, SMG$K_KEY_F16, SMG$MASK_ADR),
285 P 0360 1 ('KEY_F17', TPAS_EXIT, SMG$K_KEY_F17, SMG$MASK_ADR),
286 P 0361 1 ('KEY_F18', TPAS_EXIT, SMG$K_KEY_F18, SMG$MASK_ADR),
287 P 0362 1 ('KEY_F19', TPAS_EXIT, SMG$K_KEY_F19, SMG$MASK_ADR),
288 P 0363 1 ('KEY_F20', TPAS_EXIT, SMG$K_KEY_F20, SMG$MASK_ADR),
289 P 0364 1 ('KEY_LEFT_ARROW', TPAS_EXIT, SMG$K_KEY_LEFT_ARROW, SMG$MASK_ADR),
290 P 0365 1 ('KEY_MINUS', TPAS_EXIT, SMG$K_KEY_MINUS, SMG$MASK_ADR),
291 P 0366 1 ('KEY_PERIOD', TPAS_EXIT, SMG$K_KEY_PERIOD, SMG$MASK_ADR),
292 P 0367 1 ('KEY_PF1', TPAS_EXIT, SMG$K_KEY_PF1, SMG$MASK_ADR),
293 P 0368 1 ('KEY_PF2', TPAS_EXIT, SMG$K_KEY_PF2, SMG$MASK_ADR),
294 P 0369 1 ('KEY_PF3', TPAS_EXIT, SMG$K_KEY_PF3, SMG$MASK_ADR),
295 P 0370 1 ('KEY_PF4', TPAS_EXIT, SMG$K_KEY_PF4, SMG$MASK_ADR),
296 P 0371 1 ('KEY_RIGHT_ARROW', TPAS_EXIT, SMG$K_KEY_RIGHT_ARROW, SMG$MASK_ADR),
297 P 0372 1 ('KEY_UP_ARROW', TPAS_EXIT, SMG$K_KEY_UP_ARROW, SMG$MASK_ADR),
298 P 0373 1 ('LABEL_F1', TPAS_EXIT, SMG$K_LABEL_F1, SMG$MASK_ADR),
299 P 0374 1 ('LABEL_F2', TPAS_EXIT, SMG$K_LABEL_F2, SMG$MASK_ADR),
300 P 0375 1 ('LABEL_F3', TPAS_EXIT, SMG$K_LABEL_F3, SMG$MASK_ADR),
301 P 0376 1 ('LABEL_F4', TPAS_EXIT, SMG$K_LABEL_F4, SMG$MASK_ADR),
302 P 0377 1 ('LABEL_F5', TPAS_EXIT, SMG$K_LABEL_F5, SMG$MASK_ADR),
303 P 0378 1 ('LABEL_F6', TPAS_EXIT, SMG$K_LABEL_F6, SMG$MASK_ADR),
304 P 0379 1 ('LABEL_F7', TPAS_EXIT, SMG$K_LABEL_F7, SMG$MASK_ADR),
305 P 0380 1 ('LABEL_F8', TPAS_EXIT, SMG$K_LABEL_F8, SMG$MASK_ADR),
306 P 0381 1 ('LABEL_F9', TPAS_EXIT, SMG$K_LABEL_F9, SMG$MASK_ADR),
307 P 0382 1 ('LABEL_F10', TPAS_EXIT, SMG$K_LABEL_F10, SMG$MASK_ADR),
308 P 0383 1 ('LABEL_F11', TPAS_EXIT, SMG$K_LABEL_F11, SMG$MASK_ADR),
309 P 0384 1 ('LABEL_F12', TPAS_EXIT, SMG$K_LABEL_F12, SMG$MASK_ADR),
310 P 0385 1 ('LABEL_F13', TPAS_EXIT, SMG$K_LABEL_F13, SMG$MASK_ADR),
311 P 0386 1 ('LABEL_F14', TPAS_EXIT, SMG$K_LABEL_F14, SMG$MASK_ADR),
312 P 0387 1 ('LABEL_F15', TPAS_EXIT, SMG$K_LABEL_F15, SMG$MASK_ADR),
313 P 0388 1 ('LABEL_F16', TPAS_EXIT, SMG$K_LABEL_F16, SMG$MASK_ADR),
314 P 0389 1 ('LABEL_F17', TPAS_EXIT, SMG$K_LABEL_F17, SMG$MASK_ADR),

```



```
315 P 0390 1 ('LABEL_F18', TPAS_EXIT, , SMG$K_LABEL_F18, SMG$MASK_ADR),
316 P 0391 1 ('LABEL_F19', TPAS_EXIT, , SMG$K_LABEL_F19, SMG$MASK_ADR),
317 P 0392 1 ('LABEL_F20', TPAS_EXIT, , SMG$K_LABEL_F20, SMG$MASK_ADR),
318 P 0393 1 ('LEFT_T_CHAR', TPAS_EXIT, , SMG$K_LEFT_T_CHAR, SMG$MASK_ADR),
319 P 0394 1 ('LF_GRAPHIC', TPAS_EXIT, , SMG$K_LF_GRAPHIC, SMG$MASK_ADR),
320 P 0395 1 ('LIGHT_SCREEN', TPAS_EXIT, , SMG$K_LIGHT_SCREEN, SMG$MASK_ADR),
321 P 0396 1 ('LOWER_LEFT_CORNER', TPAS_EXIT, , SMG$K_LOWER_LEFT_CORNER, SMG$MASK_ADR),
322 P 0397 1 ('LOWER_RIGHT_CORNER', TPAS_EXIT, , SMG$K_LOWER_RIGHT_CORNER, SMG$MASK_ADR),
323 P 0398 1 ('NAME', TPAS_EXIT, , SMG$K_NAME, SMG$MASK_ADR),
324 P 0399 1 ('NEWLINE_CHAR', TPAS_EXIT, , SMG$K_NEWLINE_CHAR, SMG$MASK_ADR),
325 P 0400 1 ('PAD_CHAR', TPAS_EXIT, , SMG$K_PAD_CHAR, SMG$MASK_ADR),
326 P 0401 1 ('REQUEST_CURSOR_POSITION', TPAS_EXIT, , SMG$K_REQUEST_CURSOR_POSITION, SMG$MASK_ADR),
327 P 0402 1 ('RESTORE_CURSOR', TPAS_EXIT, , SMG$K_RESTORE_CURSOR, SMG$MASK_ADR),
328 P 0403 1 ('REVERSE_INDEX', TPAS_EXIT, , SMG$K_REVERSE_INDEX, SMG$MASK_ADR),
329 P 0404 1 ('RIGHT_T_CHAR', TPAS_EXIT, , SMG$K_RIGHT_T_CHAR, SMG$MASK_ADR),
330 P 0405 1 ('SAVE_CURSOR', TPAS_EXIT, , SMG$K_SAVE_CURSOR, SMG$MASK_ADR),
331 P 0406 1 ('SCROLL_FORWARD', TPAS_EXIT, , SMG$K_SCROLL_FORWARD, SMG$MASK_ADR),
332 P 0407 1 ('SCROLL_REVERSE', TPAS_EXIT, , SMG$K_SCROLL_REVERSE, SMG$MASK_ADR),
333 P 0408 1 ('SEL_ERASE_TO_END_DISPLAY', TPAS_EXIT, , SMG$K_SEL_ERASE_TO_END_DISPLAY, SMG$MASK_ADR),
334 P 0409 1 ('SEL_ERASE_TO_END_LINE', TPAS_EXIT, , SMG$K_SEL_ERASE_TO_END_LINE, SMG$MASK_ADR),
335 P 0410 1 ('SEL_ERASE_WHOLE_DISPLAY', TPAS_EXIT, , SMG$K_SEL_ERASE_WHOLE_DISPLAY, SMG$MASK_ADR),
336 P 0411 1 ('SEL_ERASE_WHOLE_LINE', TPAS_EXIT, , SMG$K_SEL_ERASE_WHOLE_LINE, SMG$MASK_ADR),
337 P 0412 1 ('SET_APPLICATION_KEYPAD', TPAS_EXIT, , SMG$K_SET_APPLICATION_KEYPAD, SMG$MASK_ADR),
338 P 0413 1 ('SET_CHAR_NOT_SEL_ERASE', TPAS_EXIT, , SMG$K_SET_CHAR_NOT_SEL_ERASE, SMG$MASK_ADR),
339 P 0414 1 ('SET_CHAR_SEL_ERASE', TPAS_EXIT, , SMG$K_SET_CHAR_SEL_ERASE, SMG$MASK_ADR),
340 P 0415 1 ('SET_CURSOR_ABS', TPAS_EXIT, , SMG$K_SET_CURSOR_ABS, SMG$MASK_ADR),
341 P 0416 1 ('SET_NUMERIC_KEYPAD', TPAS_EXIT, , SMG$K_SET_NUMERIC_KEYPAD, SMG$MASK_ADR),
342 P 0417 1 ('SET_SCROLL_REGION', TPAS_EXIT, , SMG$K_SET_SCROLL_REGION, SMG$MASK_ADR),
343 P 0418 1 ('SET_TAB', TPAS_EXIT, , SMG$K_SET_TAB, SMG$MASK_ADR),
344 P 0419 1 ('SINGLE_HIGH', TPAS_EXIT, , SMG$K_SINGLE_HIGH, SMG$MASK_ADR),
345 P 0420 1 ('TAB_CHAR', TPAS_EXIT, , SMG$K_TAB_CHAR, SMG$MASK_ADR),
346 P 0421 1 ('TOP_T_CHAR', TPAS_EXIT, , SMG$K_TOP_T_CHAR, SMG$MASK_ADR),
347 P 0422 1 ('TRUNCATION_ICON', TPAS_EXIT, , SMG$K_TRUNCATION_ICON, SMG$MASK_ADR),
348 P 0423 1 ('UNDERLINE_CHAR', TPAS_EXIT, , SMG$K_UNDERLINE_CHAR, SMG$MASK_ADR),
349 P 0424 1 ('UPPER_LEFT_CORNER', TPAS_EXIT, , SMG$K_UPPER_LEFT_CORNER, SMG$MASK_ADR),
350 P 0425 1 ('UPPER_RIGHT_CORNER', TPAS_EXIT, , SMG$K_UPPER_RIGHT_CORNER, SMG$MASK_ADR),
351 P 0426 1 ('VERTICAL_BAR', TPAS_EXIT, , SMG$K_VERTICAL_BAR, SMG$MASK_ADR),
352 P 0427 1 ('VT_GRAPHIC', TPAS_EXIT, , SMG$K_VT_GRAPHIC, SMG$MASK_ADR),
353 P 0428 1 ('WIDTH_NARROW', TPAS_EXIT, , SMG$K_WIDTH_NARROW, SMG$MASK_ADR),
354 P 0429 1 ('WIDTH_WIDE', TPAS_EXIT, , SMG$K_WIDTH_WIDE, SMG$MASK_ADR),
355 P 0430 1 ('PRIVATE_STR_1', TPAS_EXIT, , SMG$K_PRIVATE_STR_1, SMG$MASK_ADR),
356 P 0431 1 ('PRIVATE_STR_2', TPAS_EXIT, , SMG$K_PRIVATE_STR_2, SMG$MASK_ADR),
357 P 0432 1 ('PRIVATE_STR_3', TPAS_EXIT, , SMG$K_PRIVATE_STR_3, SMG$MASK_ADR),
358 P 0433 1 ('PRIVATE_STR_4', TPAS_EXIT, , SMG$K_PRIVATE_STR_4, SMG$MASK_ADR),
359 P 0434 1 ('PRIVATE_STR_5', TPAS_EXIT, , SMG$K_PRIVATE_STR_5, SMG$MASK_ADR),
360 P 0435 1 ('PRIVATE_STR_6', TPAS_EXIT, , SMG$K_PRIVATE_STR_6, SMG$MASK_ADR),
361 P 0436 1 ('PRIVATE_STR_7', TPAS_EXIT, , SMG$K_PRIVATE_STR_7, SMG$MASK_ADR),
362 P 0437 1 ('PRIVATE_STR_8', TPAS_EXIT, , SMG$K_PRIVATE_STR_8, SMG$MASK_ADR),
363 P 0438 1 ('PRIVATE_STR_9', TPAS_EXIT, , SMG$K_PRIVATE_STR_9, SMG$MASK_ADR),
364 P 0439 1 ('PRIVATE_STR_10', TPAS_EXIT, , SMG$K_PRIVATE_STR_10, SMG$MASK_ADR),
365 P 0440 1 (TPAS_LAMBDA, TPAS_FAIL)
366 0441 1 );
367 0442 1
368 0443 1
369 0444 1 !+ Found a name, now look for the value. Skip over the intervening
370 0445 1 ! equals sign. (If we get to '=', then we also have a capability
371 0446 1 ! name/number to remember.)
```

```

372 0447 1 !-
373 0448 1
374 P 0449 1 $STATE (EQUALS_STRING,
375 P 0450 1 ((END_OF_LINE), EQUALS_STRING, SMG$$NEXT_RECORD),
376 P 0451 1 ('='-STRING_CAP_VALUE, SMG$$STORE_CAP_MASK),
377 P 0452 1 (TPAS_SYMBOL, , SMG$$SYNTAX_ERROR),
378 P 0453 1 (TPAS_ANY, , SMG$$SYNTAX_ERROR)
379 0454 1 );
380 0455 1
381 0456 1 !+
382 0457 1 We have found a capability name, now we need to decipher its value.
383 0458 1 We found a string capability name so this better be a string value.
384 0459 1 !-
385 0460 1
386 P 0461 1 $STATE (STRING_CAP_VALUE,
387 P 0462 1 ((END_OF_LINE), STRING_CAP_VALUE, SMG$$NEXT_RECORD),
388 P 0463 1 ((STRING_CAP), BEGIN_SCAN),
389 P 0464 1 (TPAS_SYMBOL, , SMG$$SYNTAX_ERROR),
390 P 0465 1 (TPAS_ANY, , SMG$$SYNTAX_ERROR)
391 0466 1 );
392 0467 1
393 P 0468 1 $STATE (COMMA,
394 P 0469 1 ('', TPAS_EXIT, SMG$$BLANKS_OFF),
395 P 0470 1 (TPAS_BLANK, COMMA, SMG$$BLANKS_OFF),
396 P 0471 1 ((END_OF_LINE), TPAS_EXIT, SMG$$NEXT_RECORD),
397 P 0472 1 (TPAS_ANY, , SMG$$SYNTAX_ERROR)
398 0473 1 );
399 0474 1
400 0475 1 !+
401 0476 1 A string capability requires further parsing. There may be an embedded
402 0477 1 special character, an FAO call, or arithmetic. A matching quote indicates
403 0478 1 the end of the string value, which should be stored in TERMTABLE.EXE.
404 0479 1 !-
405 0480 1
406 P 0481 1 $STATE (STRING_CAP,
407 P 0482 1 (SINGLE_QUOTE, SINGLE_QUOTE_STRING, SMG$$BLANKS_ON),
408 P 0483 1 (DOUBLE_QUOTE, DOUBLE_QUOTE_STRING, SMG$$BLANKS_ON),
409 P 0484 1 (TPAS_ANY, TPAS_FAIL)
410 0485 1 );
411 0486 1
412 P 0487 1 $STATE (SINGLE_QUOTE_STRING,
413 P 0488 1 ((UNDERSCORE), SINGLE_QUOTE_STRING),
414 P 0489 1 ('$ ', SINGLE_QUOTE_STRING, CONVERT_ESCAPE),
415 P 0490 1 ((CARROT), SINGLE_QUOTE_STRING),
416 P 0491 1 ((FAO_STRING), SINGLE_QUOTE_STRING),
417 P 0492 1 ((ARITH_CAP), SINGLE_QUOTE_STRING),
418 P 0493 1 (SINGLE_QUOTE, COMMA, SMG$COPY_CAP),
419 P 0494 1 (TPAS_BLANK, SINGLE_QUOTE_STRING, SMG$$SAVE_TOKEN_STRING),
420 P 0495 1 (DOUBLE_QUOTE, , SMG$$SYNTAX_ERROR),
421 P 0496 1 (TPAS_ANY, SINGLE_QUOTE_STRING, SMG$$SAVE_TOKEN_STRING),
422 P 0497 1 (TPAS_LAMBDA, TPAS_EXIT)
423 0498 1 );
424 0499 1
425 P 0500 1 $STATE (DOUBLE_QUOTE_STRING,
426 P 0501 1 ((UNDERSCORE), DOUBLE_QUOTE_STRING),
427 P 0502 1 ('$ ', DOUBLE_QUOTE_STRING, CONVERT_ESCAPE),
428 P 0503 1 ((CARROT), DOUBLE_QUOTE_STRING),

```



```

429 P 0504 1 ((FAO STRING), DOUBLE QUOTE STRING),
430 P 0505 1 ((ARITH_CAP), DOUBLE QUOTE STRING),
431 P 0506 1 (DOUBLE_QUOTE, COMMA, SMG$COPY_CAP),
432 P 0507 1 (TPAS_BLANK, DOUBLE QUOTE STRING, SMG$SAVE_TOKEN_STRING),
433 P 0508 1 (SINGLE_QUOTE, , SMG$SYNTAX_ERROR),
434 P 0509 1 (TPAS_ANY, DOUBLE QUOTE STRING, SMG$SAVE_TOKEN_STRING),
435 P 0510 1 (TPAS_LAMBDA, TPAS_EXIT)
436 0511 1 );
437 0512 1
438 0513 1
439 0514 1 !+ Underscore is used to insert special characters into a string as
440 0515 1 normal text. If this underscore is followed by a special character,
441 0516 1 skip over it and just store the special character.
442 0517 1 !-
443 0518 1
444 P 0519 1 $STATE (UNDERScore,
445 P 0520 1 ( , UNDERSCORE2),
446 P 0521 1 (TPAS_LAMBDA, TPAS_FAIL)
447 0522 1 );
448 0523 1
449 P 0524 1 $STATE (UNDERScore2,
450 P 0525 1 ( '$', TPAS_EXIT, INSERT_DOLLAR),
451 P 0526 1 ( '^', TPAS_EXIT, INSERT_CARROT),
452 P 0527 1 ( '!', TPAS_EXIT, INSERT_EXCLAMATION),
453 P 0528 1 ( '(', TPAS_EXIT, INSERT_PARENTHESIS),
454 P 0529 1 (TPAS_LAMBDA, TPAS_FAIL)
455 0530 1 );
456 0531 1
457 0532 1 !+
458 0533 1 If we find a ^, advance to the next character and call the conversion
459 0534 1 routine. The conversion routine will complain if this is a control
460 0535 1 sequence we don't know about.
461 0536 1 !-
462 0537 1
463 P 0538 1 $STATE (CARROT,
464 P 0539 1 ( '^', CONTROL_CHAR),
465 P 0540 1 (TPAS_LAMBDA, TPAS_FAIL)
466 0541 1 );
467 0542 1
468 P 0543 1 $STATE (CONTROL_CHAR,
469 P 0544 1 (TPAS_ANY, TPAS_EXIT, CONVERT_CONTROL)
470 0545 1 );
471 0546 1
472 0547 1 !+
473 0548 1 Check for a FAO directive. Make sure it's one we understand.
474 0549 1 !-
475 P 0550 1 $STATE (FAO_STRING,
476 P 0551 1 ( '!', SMG$SAVE_TOKEN_STRING),
477 P 0552 1 (TPAS_LAMBDA, TPAS_FAIL)
478 0553 1 );
479 0554 1
480 P 0555 1 $STATE (,
481 P 0556 1 ( 'U', SMG$SAVE_TOKEN_STRING),
482 P 0557 1 (TPAS_SYMBOL, TPAS_FAIL, INVALID_DIRECTIVE),
483 P 0558 1 (TPAS_ANY, TPAS_FAIL, INVALID_DIRECTIVE)
484 0559 1 );
485 0560 1

```

```
486 P 0561 1 $STATE (,  
487 P 0562 1 ('L', TPAS_EXIT, SMG$SAVE_TOKEN_STRING, SMG$K_FAO_STRING, SMG$MASK_ADR),  
488 P 0563 1 (TPAS_SYMBOL, TPAS_FAIL, INVALID_DIRECTIVE),  
489 P 0564 1 (TPAS_ANY, TPAS_FAIL, INVALID_DIRECTIVE)  
490 0565 1 );  
491 0566 1  
492 0567 1 !+  
493 0568 1 ! Parse arithmetic string. We can't complete arithmetic processing until  
494 0569 1 ! run-time, but here we pre-compile as much as possible. Arithmetic  
495 0570 1 ! expressions are stored as number, op, number, op, number ... etc.  
496 0571 1 !-  
497 0572 1  
498 P 0573 1 $STATE (ARITH_CAP,  
499 P 0574 1 (LEFT_PAREN, ARITH_CAP2, INIT_ARITH_CAP),  
500 P 0575 1 (TPAS_BLANK, ARITH_CAP),  
501 P 0576 1 (TPAS_LAMBDA, TPAS_FAIL)  
502 0577 1 );  
503 0578 1  
504 P 0579 1 $STATE (ARITH_CAP2,  
505 P 0580 1 ((OPERAND), TPAS_EXIT), ! errors handled in OPERAND  
506 P 0581 1 (TPAS_LAMBDA, TPAS_FAIL) ! return w/out consuming next token  
507 0582 1 );  
508 0583 1  
509 P 0584 1 $STATE (OPERATOR,  
510 P 0585 1 ('+', OPERAND, , SMG$K_ADD, SMG$MASK_ADR),  
511 P 0586 1 ('-', OPERAND, , SMG$K_SUBTRACT, SMG$MASK_ADR),  
512 P 0587 1 ('*', OPERAND, , SMG$K_MULTIPLY, SMG$MASK_ADR),  
513 P 0588 1 ('/', OPERAND, , SMG$K_DIVIDE, SMG$MASK_ADR),  
514 P 0589 1 (RIGHT_PAREN, TPAS_EXIT, END_ARITH_CAP),  
515 P 0590 1 (TPAS_ANY, , EXPRESSION_ERROR)  
516 0591 1 );  
517 0592 1  
518 P 0593 1 $STATE (OPERAND,  
519 P 0594 1 (TPAS_DECIMAL, OPERATOR, STORE_OPERAND),  
520 P 0595 1 ((SUBSTITUTION), OPERATOR),  
521 P 0596 1 (TPAS_SYMBOL, , EXPRESSION_ERROR),  
522 P 0597 1 (TPAS_ANY, , EXPRESSION_ERROR)  
523 0598 1 );  
524 0599 1  
525 P 0600 1 $STATE (SUBSTITUTION,  
526 P 0601 1 ('%'),  
527 P 0602 1 (TPAS_LAMBDA, TPAS_FAIL)  
528 0603 1 );  
529 0604 1  
530 0605 1 ! % should be followed by a number  
531 P 0606 1 $STATE (  
532 P 0607 1 (TPAS_DECIMAL, TPAS_EXIT, STORE_SUBSTITUTION),  
533 P 0608 1 (TPAS_ANY, , EXPRESSION_ERROR)  
534 0609 1 );  
535 0610 1  
536 0611 1
```



```
538 0612 1 XSBTTL 'CONVERT_CONTROL - Convert ^ to control character'
539 0613 1 ROUTINE CONVERT_CONTROL =
540 0614 1
541 0615 1
542 0616 1 ++
543 0617 1 FUNCTIONAL DESCRIPTION:
544 0618 1     Stores a control character in the current capability string in
545 0619 1     TERMTABLE.EXE.
546 0620 1
547 0621 1 CALLING SEQUENCE:
548 0622 1
549 0623 1     status = CONVERT_CONTROL ()
550 0624 1
551 0625 1 FORMAL PARAMETERS:
552 0626 1
553 0627 1     NONE
554 0628 1
555 0629 1 IMPLICIT INPUTS:
556 0630 1
557 0631 1     AP     Points to TPARSE parameter block
558 0632 1
559 0633 1 IMPLICIT OUTPUTS:
560 0634 1
561 0635 1     NONE
562 0636 1
563 0637 1 COMPLETION STATUS:
564 0638 1
565 0639 1     SS$_NORMAL
566 0640 1
567 0641 1 SIDE EFFECTS:
568 0642 1
569 0643 1 --
570 0644 1
571 0645 2 BEGIN
572 0646 2 BUILTIN
573 0647 2 CALLG,
574 0648 2 AP;
575 0649 2 MAP
576 0650 2 AP : REF BLOCK [,BYTE];
577 0651 2 BIND
578 0652 2 CAP_PTRS = .AP [PARAM_L_CUR_TERM_DEF] : VECTOR [,WORD];
579 0653 2 LOCAL
580 0654 2 CONTROL_CHAR : BYTE,           ! char following ctrl
581 0655 2 MOVE_CHAR : BYTE;             ! code to place in cap string
582 0656 2
583 0657 2
584 0658 2 !+ The NAME capability should have preceeded this. Complain if it
585 0659 2 ! didn't.
586 0660 2
587 0661 2
588 0662 2 IF CAP_PTRS EQL 0
589 0663 2 THEN
590 0664 2     SIGNAL_STOP (SMG$_MISTERNAM);
591 0665 2
592 0666 2 !+
593 0667 2 ! Control must be followed by a valid character in order to constitute
594 0668 2 ! a control code. The character can be shifted or not (lower or upper
```

```
595 0669 2 | case).
596 0670 2 | Since we have already seen a control character, we can translate on
597 0671 2 | the second character in the control string.
598 0672 2 |
599 0673 2 |
600 0674 2 | CONTROL_CHAR = .(.AP [TPASL_TOKENPTR]) <0,8>;
601 0675 2 |
602 0676 2 | A control character is in the range 00 - 1F (hex) but is expressed as
603 0677 2 | 40 - 5F (hex). Strip off extra bit to get the control character.
604 0678 2 |
605 0679 2 |
606 0680 2 | MOVE_CHAR = .CONTROL_CHAR AND NOT (ZX'40');
607 0681 2 |
608 0682 2 |
609 0683 2 | We now have the byte which must be moved into the capability string.
610 0684 2 | This may not be the first byte of the string - check for saved characters
611 0685 2 | which have not been copied yet, and append in the control code.
612 0686 2 |
613 0687 2 |
614 0688 2 | CALLG (.AP, SMG$FLUSH_SAVED_BUFFER); ! copy saved string
615 0689 2 |
616 0690 2 | AP [PARAM_L_SAVED_TOKENCNT] = 1;
617 0691 2 | AP [PARAM_L_SAVED_TOKENSTR] = MOVE_CHAR; ! control char is now 'saved'
618 0692 2 |
619 0693 2 | CALLG (.AP, SMG$FLUSH_SAVED_BUFFER); ! append control char
620 0694 2 |
621 0695 2 | RETURN (SS$NORMAL);
622 0696 2 |
623 0697 1 | END; ! end of routine CONVERT_CONTROL
```

```
.TITLE SMG$STRING_TABLES TPARSE tables for string capa
.IDENT \1-003\ bilities
.PSECT _LIB$KEY1$,NOWRT, SHR, PIC,1
```

```
00000 :TPASKEYSTO
44 4E 45 00000 U.29: .BLKB 0
:TPASKEYST
FF 00003 U.31: .ASCII \END\
00004 :TPASKEYSTO .BYTE -1
4E 41 45 4C 4F 4F 42 00004 U.34: .BLKB 0
:TPASKEYST
FF 0000B U.36: .ASCII \BOOLEAN\
0000C :TPASKEYSTO .BYTE -1
43 49 52 45 4D 55 4E 0000C U.39: .BLKB 0
:TPASKEYST
FF 00013 U.41: .ASCII \NUMERIC\
00014 :TPASKEYSTO .BYTE -1
47 4E 49 52 54 53 00014 U.44: .BLKB 0
:TPASKEYST
FF 0001A U.46: .ASCII \STRING\
.BYTE -1
```


[illegible]

Address	Hex Data	Assembly	Comment
52 4F 43 53 52 45 44 4E 55 5F 4E 49 47 45 42	000B6	:TPASKEYSTO	
	U.120: .BLKB		0
	000B6	:TPASKEYST	
	U.122: .ASCII		\BEGIN_UNDERSCORE\
	45 000C5	:TPASKEYSTO	
	FF 000C6	:TPASKEYST	
	000C7	:TPASKEYSTO	
52 41 48 43 5F 54 5F 4D 4F 54 54 4F 42	000C7	:TPASKEYST	
	U.127: .BLKB		0
	FF 000D4	:TPASKEYSTO	
	000D5	:TPASKEYST	
	U.129: .ASCII		\BOTTOM_T_CHAR\
	FF 000D4	:TPASKEYSTO	
	000D5	:TPASKEYST	
	U.134: .BLKB		0
42 41 54 5F 52 41 45 4C 43	000D5	:TPASKEYST	
	U.136: .ASCII		\CLEAR_TAB\
	FF 000DE	:TPASKEYSTO	
	000DF	:TPASKEYST	
	U.141: .BLKB		0
43 49 48 50 41 52 47 5F 52 43	000DF	:TPASKEYST	
	U.143: .ASCII		\CR_GRAPHIC\
	FF 000E9	:TPASKEYSTO	
	000EA	:TPASKEYST	
	U.148: .BLKB		0
52 41 48 43 5F 53 53 4F 52 43	000EA	:TPASKEYST	
	U.150: .ASCII		\CROSS_CHAR\
	FF 000F4	:TPASKEYSTO	
	000F5	:TPASKEYST	
	U.155: .BLKB		0
4E 57 4F 44 5F 52 4F 53 52 55 43	000F5	:TPASKEYST	
	U.157: .ASCII		\CURSOR_DOWN\
	FF 00100	:TPASKEYSTO	
	00101	:TPASKEYST	
	U.162: .BLKB		0
54 46 45 4C 5F 52 4F 53 52 55 43	00101	:TPASKEYST	
	U.164: .ASCII		\CURSOR_LEFT\
	FF 0010C	:TPASKEYSTO	
	0010D	:TPASKEYST	
	U.169: .BLKB		0
4E 49 4C 5F 54 58 45 4E 5F 52 4F 53 52 55 43	0010D	:TPASKEYST	
	U.171: .ASCII		\CURSOR_NEXT_LINE\
	45 0011C	:TPASKEYSTO	
	FF 0011D	:TPASKEYST	
	0011E	:TPASKEYSTO	
	U.176: .BLKB		0
4E 4F 49 54 49 53 4F 50 5F 52 4F 53 52 55 43	0011E	:TPASKEYST	
	U.178: .ASCII		\CURSOR_POSITION_REPORT\
	54 52 4F 50 45 52 5F 0012D	:TPASKEYSTO	
	FF 00134	:TPASKEYST	
	00135	:TPASKEYSTO	
	U.183: .BLKB		0
49 44 45 45 43 45 52 50 5F 52 4F 53 52 55 43	00135	:TPASKEYST	
	U.185: .ASCII		\CURSOR_PRECEEDING_LINE\
	45 4E 49 4C 5F 47 4E 00144	:TPASKEYSTO	
	FF 0014B	:TPASKEYST	
	0014C	:TPASKEYSTO	
	U.190: .BLKB		0
54 48 47 49 52 5F 52 4F 53 52 55 43	0014C	:TPASKEYST	

Address	Hex	ASCII	Comment
50	55	5F	52
4F	53	52	55
43	00158	U.192: .ASCII	\CURSOR_RIGHT\
FF	00159	:TPASKEYSTO	-1
43	00159	U.197: .BLKB	0
FF	00162	:TPASKEYSTO	-1
4E	45	45	52
43	53	5F	4B
52	41	44	00163
44	00163	U.204: .BLKB	0
FF	0016E	:TPASKEYSTO	-1
52	41	48	43
5F	45	54	45
4C	45	44	0016F
44	0016F	U.211: .BLKB	0
FF	0017A	:TPASKEYSTO	-1
45	4E	49	4C
5F	45	54	45
4C	45	44	0017B
44	0017B	U.218: .BLKB	0
FF	00186	:TPASKEYSTO	-1
54	55	42	49
52	54	54	41
5F	45	43	49
56	45	44	00187
44	00187	U.225: .BLKB	0
53	45	00196	U.227: .ASCII
FF	00198	:TPASKEYSTO	-1
54	4F	42	5F
48	47	49	48
5F	45	4C	42
55	4F	44	00199
44	00199	U.232: .BLKB	0
4D	4F	54	001A8
FF	001AB	:TPASKEYSTO	-1
50	4F	54	5F
48	47	49	48
5F	45	4C	42
55	4F	44	001AC
44	001AC	U.239: .BLKB	0
FF	001BB	:TPASKEYSTO	-1
45	44	49	57
5F	45	4C	42
55	4F	44	001BC
44	001BC	U.246: .BLKB	0
FF	001C7	:TPASKEYSTO	-1
45	54	41	43
49	4C	50	55
44	001C8	U.253: .BLKB	0
FF	001D1	:TPASKEYSTO	-1
43	5F	45	54
41	4E	52	45
54	4C	41	5F
44	4E	45	001D2
45	001D2	U.260: .BLKB	0
52	41	48	001E1
FF	001E4	:TPASKEYSTO	-1
52	41	48	001E5
FF	001E5	U.267: .BLKB	0

Hex	ASCII	Function
4B 4E 49 4C 42 5F 44 4E 45	001E5	:TPASKEYSTO
	FF 001EE	U.269: .ASCII BYTE
	001EF	:TPASKEYSTO
44 4C 4F 42 5F 44 4E 45	001EF	U.274: .BLKB
	FF 001F7	U.276: .ASCII BYTE
	001F8	:TPASKEYSTO
45 44 4F 4D 5F 45 54 45 4C 45 44 5F 44 4E 45	001F8	U.281: .BLKB
	FF 00207	U.283: .ASCII BYTE
	00208	:TPASKEYSTO
45 44 4F 4D 5F 54 52 45 53 4E 49 5F 44 4E 45	00208	U.288: .BLKB
	FF 00217	U.290: .ASCII BYTE
	00218	:TPASKEYSTO
4E 49 57 41 52 44 5F 45 4E 49 4C 5F 44 4E 45	00218	U.295: .BLKB
	FF 00227	U.297: .ASCII BYTE
	0022D	:TPASKEYSTO
	0022E	U.302: .BLKB
45 53 52 45 56 45 52 5F 44 4E 45	0022E	:TPASKEYSTO
	FF 00239	U.304: .ASCII BYTE
	0023A	:TPASKEYSTO
45 52 4F 43 53 52 45 44 4E 55 5F 44 4E 45	0023A	U.309: .BLKB
	FF 00248	U.311: .ASCII BYTE
	00249	:TPASKEYSTO
49 44 5F 44 4E 45 5F 4F 54 5F 45 53 41 52 45	00249	U.316: .BLKB
	FF 00258	U.318: .ASCII BYTE
	0025D	:TPASKEYSTO
	0025E	U.323: .BLKB
49 4C 5F 44 4E 45 5F 4F 54 5F 45 53 41 52 45	0025E	:TPASKEYSTO
	FF 0026D	U.325: .ASCII BYTE
	0026F	:TPASKEYSTO
	00270	U.330: .BLKB
53 49 44 5F 45 4C 4F 48 57 5F 45 53 41 52 45	00270	:TPASKEYSTO
	FF 0027F	U.332: .ASCII BYTE
	00283	:TPASKEYSTO
	00284	U.337: .BLKB
4E 49 4C 5F 45 4C 4F 48 57 5F 45 53 41 52 45	00284	:TPASKEYSTO
	FF 00293	U.339: .ASCII

Offset	Hex	ASCII	Comment
00294	FF	:TPASKEYSTO	00294
00295	U.344:	.BLKB	0
00295	FF	:TPASKEYSTO	00295
0029F	U.346:	.ASCII	\FF_GRAPHIC\
002A0	FF	:TPASKEYSTO	002A0
002A0	U.351:	.BLKB	0
002A0	FF	:TPASKEYSTO	002A0
002A4	U.353:	.ASCII	\HOME\
002A5	FF	:TPASKEYSTO	002A5
002A5	U.358:	.BLKB	0
002A5	FF	:TPASKEYSTO	002A5
002B3	U.360:	.ASCII	\HORIZONTAL_BAR\
002B4	FF	:TPASKEYSTO	002B4
002B4	U.365:	.BLKB	0
002B4	FF	:TPASKEYSTO	002B4
002BE	U.367:	.ASCII	\HT_GRAPHIC\
002BF	FF	:TPASKEYSTO	002BF
002BF	U.372:	.BLKB	0
002BF	FF	:TPASKEYSTO	002BF
002C4	U.374:	.ASCII	\INDEX\
002C5	FF	:TPASKEYSTO	002C5
002C5	U.379:	.BLKB	0
002C5	FF	:TPASKEYSTO	002C5
002D0	U.381:	.ASCII	\INIT_STRING\
002D1	FF	:TPASKEYSTO	002D1
002D1	U.386:	.BLKB	0
002D1	FF	:TPASKEYSTO	002D1
002DC	U.388:	.ASCII	\INSERT_CHAR\
002DD	FF	:TPASKEYSTO	002DD
002DD	U.393:	.BLKB	0
002DD	FF	:TPASKEYSTO	002DD
002E8	U.395:	.ASCII	\INSERT_LINE\
002E9	FF	:TPASKEYSTO	002E9
002E9	U.400:	.BLKB	0
002E9	FF	:TPASKEYSTO	002E9
002F3	U.402:	.ASCII	\INSERT_PAD\
002F4	FF	:TPASKEYSTO	002F4
002F4	U.407:	.BLKB	0
002F4	FF	:TPASKEYSTO	002F4
002F9	U.409:	.ASCII	\KEY_0\
002FA	FF	:TPASKEYSTO	002FA
002FA	U.414:	.BLKB	0
002FA	FF	:TPASKEYSTO	002FA
002FF	U.416:	.ASCII	\KEY_1\
00300	FF	:TPASKEYSTO	00300

Line	Address	Hex	ASCII	Comment
32	5F 59 45 4B	00300	U.421: .BLKB :TPASKEYST	0
		FF 00305	U.423: .ASCII .BYTE	\KEY_2\ -1
		00306	:TPASKEYSTO	
33	5F 59 45 4B	00306	U.428: .BLKB :TPASKEYST	0
		FF 0030B	U.430: .ASCII .BYTE	\KEY_3\ -1
		0030C	:TPASKEYSTO	
34	5F 59 45 4B	0030C	U.435: .BLKB :TPASKEYST	0
		FF 00311	U.437: .ASCII .BYTE	\KEY_4\ -1
		00312	:TPASKEYSTO	
35	5F 59 45 4B	00312	U.442: .BLKB :TPASKEYST	0
		FF 00317	U.444: .ASCII .BYTE	\KEY_5\ -1
		00318	:TPASKEYSTO	
36	5F 59 45 4B	00318	U.449: .BLKB :TPASKEYST	0
		FF 0031D	U.451: .ASCII .BYTE	\KEY_6\ -1
		0031E	:TPASKEYSTO	
37	5F 59 45 4B	0031E	U.456: .BLKB :TPASKEYST	0
		FF 00323	U.458: .ASCII .BYTE	\KEY_7\ -1
		00324	:TPASKEYSTO	
38	5F 59 45 4B	00324	U.463: .BLKB :TPASKEYST	0
		FF 00329	U.465: .ASCII .BYTE	\KEY_8\ -1
		0032A	:TPASKEYSTO	
39	5F 59 45 4B	0032A	U.470: .BLKB :TPASKEYST	0
		FF 0032F	U.472: .ASCII .BYTE	\KEY_9\ -1
		00330	:TPASKEYSTO	
45	43 41 50 53 4B 43 41 42 5F 59 45 4B	00330	U.477: .BLKB :TPASKEYST	0
		FF 0033D	U.479: .ASCII .BYTE	\KEY_BACKSPACE\ -1
		0033E	:TPASKEYSTO	
	41 4D 4D 4F 43 5F 59 45 4B	0033E	U.484: .BLKB :TPASKEYST	0
		FF 00347	U.486: .ASCII .BYTE	\KEY_COMMA\ -1
		00348	:TPASKEYSTO	
57	4F 52 52 41 5F 4E 57 4F 44 5F 59 45 4B	00348	U.491: .BLKB :TPASKEYST	0
		FF 00356	U.493: .ASCII .BYTE	\KEY_DOWN_ARROW\ -1
		00357	:TPASKEYSTO	
	31 45 5F 59 45 4B	00357	U.498: .BLKB :TPASKEYST	0

Line	Address	Hex	ASCII	Comment
	0035D	FF	U.500: .ASCII	\KEY_E1\
	0035E		:TPASKEYSTO	-1
32	45	5F	59	45
	0035E	4B	U.505: .BLKB	0
			:TPASKEYST	
	00364	FF	U.507: .ASCII	\KEY_E2\
	00365		:TPASKEYSTO	-1
33	45	5F	59	45
	00365	4B	U.512: .BLKB	0
			:TPASKEYST	
	0036B	FF	U.514: .ASCII	\KEY_E3\
	0036C		:TPASKEYSTO	-1
34	45	5F	59	45
	0036C	4B	U.519: .BLKB	0
			:TPASKEYST	
	00372	FF	U.521: .ASCII	\KEY_E4\
	00373		:TPASKEYSTO	-1
35	45	5F	59	45
	00373	4B	U.526: .BLKB	0
			:TPASKEYST	
	00379	FF	U.528: .ASCII	\KEY_E5\
	0037A		:TPASKEYSTO	-1
36	45	5F	59	45
	0037A	4B	U.533: .BLKB	0
			:TPASKEYST	
	00380	FF	U.535: .ASCII	\KEY_E6\
	00381		:TPASKEYSTO	-1
52	45	54	4E	45
	00381	4B	U.540: .BLKB	0
			:TPASKEYST	
	0038A	FF	U.542: .ASCII	\KEY_ENTER\
	0038B		:TPASKEYSTO	-1
31	46	5F	59	45
	0038B	4B	U.547: .BLKB	0
			:TPASKEYST	
	00391	FF	U.549: .ASCII	\KEY_F1\
	00392		:TPASKEYSTO	-1
32	46	5F	59	45
	00392	4B	U.554: .BLKB	0
			:TPASKEYST	
	00398	FF	U.556: .ASCII	\KEY_F2\
	00399		:TPASKEYSTO	-1
33	46	5F	59	45
	00399	4B	U.561: .BLKB	0
			:TPASKEYST	
	0039F	FF	U.563: .ASCII	\KEY_F3\
	003A0		:TPASKEYSTO	-1
34	46	5F	59	45
	003A0	4B	U.568: .BLKB	0
			:TPASKEYST	
	003A6	FF	U.570: .ASCII	\KEY_F4\
	003A7		:TPASKEYSTO	-1
35	46	5F	59	45
	003A7	4B	U.575: .BLKB	0
			:TPASKEYST	
	003AD	FF	U.577: .ASCII	\KEY_F5\
			:TPASKEYSTO	-1

						003AE	:TPASKEYSTO	
							U.582: .BLKB	0
36	46	5F	59	45	4B	003AE	:TPASKEYST	
					FF	003B4	U.584: .ASCII	\KEY_F6\
							.BYTE	-1
						003B5	:TPASKEYSTO	
							U.589: .BLKB	0
37	46	5F	59	45	4B	003B5	:TPASKEYST	
					FF	003BB	U.591: .ASCII	\KEY_F7\
							.BYTE	-1
						003BC	:TPASKEYSTO	
							U.596: .BLKB	0
38	46	5F	59	45	4B	003BC	:TPASKEYST	
					FF	003C2	U.598: .ASCII	\KEY_F8\
							.BYTE	-1
						003C3	:TPASKEYSTO	
							U.603: .BLKB	0
39	46	5F	59	45	4B	003C3	:TPASKEYST	
					FF	003C9	U.605: .ASCII	\KEY_F9\
							.BYTE	-1
						003CA	:TPASKEYSTO	
							U.610: .BLKB	0
30	31	46	5F	59	45	4B	003CA	:TPASKEYST
					FF	003D1	U.612: .ASCII	\KEY_F10\
							.BYTE	-1
						003D2	:TPASKEYSTO	
							U.617: .BLKB	0
31	31	46	5F	59	45	4B	003D2	:TPASKEYST
					FF	003D9	U.619: .ASCII	\KEY_F11\
							.BYTE	-1
						003DA	:TPASKEYSTO	
							U.624: .BLKB	0
32	31	46	5F	59	45	4B	003DA	:TPASKEYST
					FF	003E1	U.626: .ASCII	\KEY_F12\
							.BYTE	-1
						003E2	:TPASKEYSTO	
							U.631: .BLKB	0
33	31	46	5F	59	45	4B	003E2	:TPASKEYST
					FF	003E9	U.633: .ASCII	\KEY_F13\
							.BYTE	-1
						003EA	:TPASKEYSTO	
							U.638: .BLKB	0
34	31	46	5F	59	45	4B	003EA	:TPASKEYST
					FF	003F1	U.640: .ASCII	\KEY_F14\
							.BYTE	-1
						003F2	:TPASKEYSTO	
							U.645: .BLKB	0
35	31	46	5F	59	45	4B	003F2	:TPASKEYST
					FF	003F9	U.647: .ASCII	\KEY_F15\
							.BYTE	-1
						003FA	:TPASKEYSTO	
							U.652: .BLKB	0
36	31	46	5F	59	45	4B	003FA	:TPASKEYST
					FF	00401	U.654: .ASCII	\KEY_F16\
							.BYTE	-1
						00402	:TPASKEYSTO	
							U.659: .BLKB	0

Address	Hex	ASCII	Comment
37 31 46 5F 59 45 4B	00402	:TPASKEYST	
	FF 00409	U.661: .ASCII	\KEY_F17\
	0040A	:TPASKEYSTO	
		U.666: .BLKB	0
38 31 46 5F 59 45 4B	0040A	:TPASKEYST	
	FF 00411	U.668: .ASCII	\KEY_F18\
	00412	:TPASKEYSTO	
		U.673: .BLKB	0
39 31 46 5F 59 45 4B	00412	:TPASKEYST	
	FF 00419	U.675: .ASCII	\KEY_F19\
	0041A	:TPASKEYSTO	
		U.680: .BLKB	0
30 32 46 5F 59 45 4B	0041A	:TPASKEYST	
	FF 00421	U.682: .ASCII	\KEY_F20\
	00422	:TPASKEYSTO	
		U.687: .BLKB	0
57 4F 52 52 41 5F 54 46 45 4C 5F 59 45 4B	00422	:TPASKEYST	
	FF 00430	U.689: .ASCII	\KEY_LEFT_ARROW\
	00431	:TPASKEYSTO	
		U.694: .BLKB	0
53 55 4E 49 4D 5F 59 45 4B	00431	:TPASKEYST	
	FF 0043A	U.696: .ASCII	\KEY_MINUS\
	0043B	:TPASKEYSTO	
		U.701: .BLKB	0
44 4F 49 52 45 50 5F 59 45 4B	0043B	:TPASKEYST	
	FF 00445	U.703: .ASCII	\KEY_PERIOD\
	00446	:TPASKEYSTO	
		U.708: .BLKB	0
31 46 50 5F 59 45 4B	00446	:TPASKEYST	
	FF 0044D	U.710: .ASCII	\KEY_PF1\
	0044E	:TPASKEYSTO	
		U.715: .BLKB	0
32 46 50 5F 59 45 4B	0044E	:TPASKEYST	
	FF 00455	U.717: .ASCII	\KEY_PF2\
	00456	:TPASKEYSTO	
		U.722: .BLKB	0
33 46 50 5F 59 45 4B	00456	:TPASKEYST	
	FF 0045D	U.724: .ASCII	\KEY_PF3\
	0045E	:TPASKEYSTO	
		U.729: .BLKB	0
34 46 50 5F 59 45 4B	0045E	:TPASKEYST	
	FF 00465	U.731: .ASCII	\KEY_PF4\
	00466	:TPASKEYSTO	
		U.736: .BLKB	0
57 4F 52 52 41 5F 54 48 47 49 52 5F 59 45 4B	00466	:TPASKEYST	
		U.738: .ASCII	\KEY_RIGHT_ARROW\

											FF	00475	:TPASKEYSTO	.BYTE	-1	:
												00476	:TPASKEYSTO	.BLKB	0	:
57	4F	52	52	41	5F	50	55	5F	59	45	4B	00476	:TPASKEYSTO	.ASCII	\KEY_UP_ARROW\	:
													U.743:	.BLKB	0	:
													U.745:	.ASCII	\KEY_UP_ARROW\	:
											FF	00482	:TPASKEYSTO	.BYTE	-1	:
												00483	:TPASKEYSTO	.BLKB	0	:
													U.750:	.BLKB	0	:
31	46	5F	4C	45	42	41	4C	00483	:TPASKEYSTO	.ASCII	\LABEL_F1\	:				
													U.752:	.ASCII	\LABEL_F1\	:
											FF	0048B	:TPASKEYSTO	.BYTE	-1	:
												0048C	:TPASKEYSTO	.BLKB	0	:
													U.757:	.BLKB	0	:
32	46	5F	4C	45	42	41	4C	0048C	:TPASKEYSTO	.ASCII	\LABEL_F2\	:				
													U.759:	.ASCII	\LABEL_F2\	:
											FF	00494	:TPASKEYSTO	.BYTE	-1	:
												00495	:TPASKEYSTO	.BLKB	0	:
													U.764:	.BLKB	0	:
33	46	5F	4C	45	42	41	4C	00495	:TPASKEYSTO	.ASCII	\LABEL_F3\	:				
													U.766:	.ASCII	\LABEL_F3\	:
											FF	0049D	:TPASKEYSTO	.BYTE	-1	:
												0049E	:TPASKEYSTO	.BLKB	0	:
													U.771:	.BLKB	0	:
34	46	5F	4C	45	42	41	4C	0049E	:TPASKEYSTO	.ASCII	\LABEL_F4\	:				
													U.773:	.ASCII	\LABEL_F4\	:
											FF	004A6	:TPASKEYSTO	.BYTE	-1	:
												004A7	:TPASKEYSTO	.BLKB	0	:
													U.778:	.BLKB	0	:
35	46	5F	4C	45	42	41	4C	004A7	:TPASKEYSTO	.ASCII	\LABEL_F5\	:				
													U.780:	.ASCII	\LABEL_F5\	:
											FF	004AF	:TPASKEYSTO	.BYTE	-1	:
												004B0	:TPASKEYSTO	.BLKB	0	:
													U.785:	.BLKB	0	:
36	46	5F	4C	45	42	41	4C	004B0	:TPASKEYSTO	.ASCII	\LABEL_F6\	:				
													U.787:	.ASCII	\LABEL_F6\	:
											FF	004B8	:TPASKEYSTO	.BYTE	-1	:
												004B9	:TPASKEYSTO	.BLKB	0	:
													U.792:	.BLKB	0	:
37	46	5F	4C	45	42	41	4C	004B9	:TPASKEYSTO	.ASCII	\LABEL_F7\	:				
													U.794:	.ASCII	\LABEL_F7\	:
											FF	004C1	:TPASKEYSTO	.BYTE	-1	:
												004C2	:TPASKEYSTO	.BLKB	0	:
													U.799:	.BLKB	0	:
38	46	5F	4C	45	42	41	4C	004C2	:TPASKEYSTO	.ASCII	\LABEL_F8\	:				
													U.801:	.ASCII	\LABEL_F8\	:
											FF	004CA	:TPASKEYSTO	.BYTE	-1	:
												004CB	:TPASKEYSTO	.BLKB	0	:
													U.806:	.BLKB	0	:
39	46	5F	4C	45	42	41	4C	004CB	:TPASKEYSTO	.ASCII	\LABEL_F9\	:				
													U.808:	.ASCII	\LABEL_F9\	:
											FF	004D3	:TPASKEYSTO	.BYTE	-1	:
												004D4	:TPASKEYSTO	.BLKB	0	:
													U.813:	.BLKB	0	:
30	31	46	5F	4C	45	42	41	4C	004D4	:TPASKEYSTO	.ASCII	\LABEL_F10\	:			
													U.815:	.ASCII	\LABEL_F10\	:
											FF	004DD	:TPASKEYSTO	.BYTE	-1	:
												004DE	:TPASKEYSTO			:

31	31	46	5F	4C	45	42	41	4C	004DE	U.820: .BLKB	0		
										:TPASKEYST			
								FF	004E7	U.822: .ASCII	\LABEL_F11\	:	
										.BYTE	-1	:	
									004E8	:TPASKEYSTO		:	
32	31	46	5F	4C	45	42	41	4C	004E8	U.827: .BLKB	0		
										:TPASKEYST			
								FF	004F1	U.829: .ASCII	\LABEL_F12\	:	
										.BYTE	-1	:	
									004F2	:TPASKEYSTO		:	
33	31	46	5F	4C	45	42	41	4C	004F2	U.834: .BLKB	0		
										:TPASKEYST			
								FF	004FB	U.836: .ASCII	\LABEL_F13\	:	
										.BYTE	-1	:	
									004FC	:TPASKEYSTO		:	
34	31	46	5F	4C	45	42	41	4C	004FC	U.841: .BLKB	0		
										:TPASKEYST			
								FF	00505	U.843: .ASCII	\LABEL_F14\	:	
										.BYTE	-1	:	
									00506	:TPASKEYSTO		:	
35	31	46	5F	4C	45	42	41	4C	00506	U.848: .BLKB	0		
										:TPASKEYST			
								FF	0050F	U.850: .ASCII	\LABEL_F15\	:	
										.BYTE	-1	:	
									00510	:TPASKEYSTO		:	
36	31	46	5F	4C	45	42	41	4C	00510	U.855: .BLKB	0		
										:TPASKEYST			
								FF	00519	U.857: .ASCII	\LABEL_F16\	:	
										.BYTE	-1	:	
									0051A	:TPASKEYSTO		:	
37	31	46	5F	4C	45	42	41	4C	0051A	U.862: .BLKB	0		
										:TPASKEYST			
								FF	00523	U.864: .ASCII	\LABEL_F17\	:	
										.BYTE	-1	:	
									00524	:TPASKEYSTO		:	
38	31	46	5F	4C	45	42	41	4C	00524	U.869: .BLKB	0		
										:TPASKEYST			
								FF	0052D	U.871: .ASCII	\LABEL_F18\	:	
										.BYTE	-1	:	
									0052E	:TPASKEYSTO		:	
39	31	46	5F	4C	45	42	41	4C	0052E	U.876: .BLKB	0		
										:TPASKEYST			
								FF	00537	U.878: .ASCII	\LABEL_F19\	:	
										.BYTE	-1	:	
									00538	:TPASKEYSTO		:	
30	32	46	5F	4C	45	42	41	4C	00538	U.883: .BLKB	0		
										:TPASKEYST			
								FF	00541	U.885: .ASCII	\LABEL_F20\	:	
										.BYTE	-1	:	
									00542	:TPASKEYSTO		:	
52	41	48	43	5F	54	5F	54	46	45	4C	00542	U.890: .BLKB	0
											:TPASKEYST		
										FF	0054D	U.892: .ASCII	\LEFT_T_CHAR\
												.BYTE	-1
										0054E	:TPASKEYSTO		:
43	49	48	50	41	52	47	5F	46	4C	0054E	U.897: .BLKB	0	
											:TPASKEYST		

Address	Hex	ASCII	Comment
00558	FF	U.899: .ASCII	\LF_GRAPHIC\
00559	FF	:TPASKEYSTO	-1
00559	4C	U.904: .BLKB	0
00559	FF	:TPASKEYST	0
00565	FF	U.906: .ASCII	\LIGHT_SCREEN\
00566	FF	:TPASKEYSTO	-1
00566	4C	U.911: .BLKB	0
00566	FF	:TPASKEYST	0
00575	52	U.913: .ASCII	\LOWER_LEFT_CORNER\
00577	FF	:TPASKEYSTO	-1
00578	FF	U.918: .BLKB	0
00578	4C	:TPASKEYST	0
00587	52	U.920: .ASCII	\LOWER_RIGHT_CORNER\
0058A	FF	:TPASKEYSTO	-1
0058B	FF	U.925: .BLKB	0
0058B	4E	:TPASKEYST	0
0058F	FF	U.927: .ASCII	\NAME\
00590	FF	:TPASKEYSTO	-1
00590	4E	U.932: .BLKB	0
00590	FF	:TPASKEYST	0
0059C	FF	U.934: .ASCII	\NEWLINE_CHAR\
0059D	FF	:TPASKEYSTO	-1
0059D	50	U.939: .BLKB	0
0059D	FF	:TPASKEYST	0
005A5	FF	U.941: .ASCII	\PAD_CHAR\
005A6	FF	:TPASKEYSTO	-1
005A6	52	U.946: .BLKB	0
005A6	FF	:TPASKEYST	0
005B5	50	U.948: .ASCII	\REQUEST_CURSOR_POSITION\
005BD	FF	:TPASKEYSTO	-1
005BE	FF	U.953: .BLKB	0
005BE	52	:TPASKEYST	0
005CC	FF	U.955: .ASCII	\RESTORE_CURSOR\
005CD	FF	:TPASKEYSTO	-1
005CD	52	U.960: .BLKB	0
005CD	FF	:TPASKEYST	0
005DA	FF	U.962: .ASCII	\REVERSE_INDEX\
005DB	FF	:TPASKEYSTO	-1
005DB	52	U.967: .BLKB	0
005DB	FF	:TPASKEYST	0
005E7	FF	U.969: .ASCII	\RIGHT_T_CHAR\
005E8	FF	:TPASKEYSTO	-1
005E8	52	U.974: .BLKB	0

Hex	ASCII	Function
52 4F 53 52 55 43 5F 45 56 41 53	005E8	:TPASKEYSTO
	FF 005F3	U.976: .ASCII
	005F4	:TPASKEYSTO
44 52 41 57 52 4F 46 5F 4C 4C 4F 52 43 53	005F4	U.981: .BLKB
	FF 00602	:TPASKEYSTO
	00603	U.983: .ASCII
45 53 52 45 56 45 52 5F 4C 4C 4F 52 43 53	00603	:TPASKEYSTO
	FF 00611	U.990: .ASCII
	00612	:TPASKEYSTO
4E 45 5F 4F 54 5F 45 53 41 52 45 5F 4C 45 53	00612	U.995: .BLKB
	FF 00621	:TPASKEYSTO
	0062A	U.997: .ASCII
	0062B	:TPASKEYSTO
4E 45 5F 4F 54 5F 45 53 41 52 45 5F 4C 45 53	0062B	U.1002: .BLKB
	FF 0063A	:TPASKEYSTO
	00640	U.1004: .ASCII
	00641	:TPASKEYSTO
45 4C 4F 48 57 5F 45 53 41 52 45 5F 4C 45 53	00641	U.1009: .BLKB
	FF 00650	:TPASKEYSTO
	00658	U.1011: .ASCII
	00659	:TPASKEYSTO
45 4C 4F 48 57 5F 45 53 41 52 45 5F 4C 45 53	00659	U.1016: .BLKB
	FF 00668	:TPASKEYSTO
	0066D	U.1018: .ASCII
	0066E	:TPASKEYSTO
4E 4F 49 54 41 43 49 4C 50 50 41 5F 54 45 53	0066E	U.1023: .BLKB
	FF 0067D	:TPASKEYSTO
	00684	U.1025: .ASCII
	00685	:TPASKEYSTO
45 53 5F 54 4F 4E 5F 52 41 48 43 5F 54 45 53	00685	U.1030: .BLKB
	FF 00694	:TPASKEYSTO
	0069B	U.1032: .ASCII
	0069C	:TPASKEYSTO
52 45 5F 4C 45 53 5F 52 41 48 43 5F 54 45 53	0069C	U.1037: .BLKB
	FF 006AB	:TPASKEYSTO
	006AE	U.1039: .ASCII
	006AF	:TPASKEYSTO
		U.1044: .BLKB

[illegible]

												0074F	:TPASKEYSTO		
												U.1121: .BLKB	0		
52	41	42	5F	4C	41	43	49	54	52	45	56	0074F	:TPASKEYST		
												U.1123: .ASCII	\VERTICAL_BAR\	:	
											FF	0075B	:TPASKEYST		
												U.1128: .BLKB	0		
												0075C	:TPASKEYST		
												U.1130: .ASCII	\VT_GRAPHIC\	:	
											FF	00766	:TPASKEYST		
												U.1135: .BLKB	0		
												00767	:TPASKEYST		
57	4F	52	52	41	4E	5F	48	54	44	49	57	00767	:TPASKEYST		
												U.1137: .ASCII	\WIDTH_NARROW\	:	
											FF	00773	:TPASKEYST		
												U.1142: .BLKB	0		
												00774	:TPASKEYST		
												U.1144: .ASCII	\WIDTH_WIDE\	:	
											FF	0077E	:TPASKEYST		
												U.1149: .BLKB	0		
												0077F	:TPASKEYST		
31	5F	52	54	53	5F	45	54	41	56	49	50	0077F	:TPASKEYST		
												U.1151: .ASCII	\PRIVATE_STR_1\	:	
											FF	0078C	:TPASKEYST		
												U.1156: .BLKB	0		
												0078D	:TPASKEYST		
32	5F	52	54	53	5F	45	54	41	56	49	50	0078D	:TPASKEYST		
												U.1158: .ASCII	\PRIVATE_STR_2\	:	
											FF	0079A	:TPASKEYST		
												U.1163: .BLKB	0		
												0079B	:TPASKEYST		
33	5F	52	54	53	5F	45	54	41	56	49	50	0079B	:TPASKEYST		
												U.1165: .ASCII	\PRIVATE_STR_3\	:	
											FF	007A8	:TPASKEYST		
												U.1170: .BLKB	0		
												007A9	:TPASKEYST		
34	5F	52	54	53	5F	45	54	41	56	49	50	007A9	:TPASKEYST		
												U.1172: .ASCII	\PRIVATE_STR_4\	:	
											FF	007B6	:TPASKEYST		
												U.1177: .BLKB	0		
												007B7	:TPASKEYST		
35	5F	52	54	53	5F	45	54	41	56	49	50	007B7	:TPASKEYST		
												U.1179: .ASCII	\PRIVATE_STR_5\	:	
											FF	007C4	:TPASKEYST		
												U.1184: .BLKB	0		
												007C5	:TPASKEYST		
36	5F	52	54	53	5F	45	54	41	56	49	50	007C5	:TPASKEYST		
												U.1186: .ASCII	\PRIVATE_STR_6\	:	
											FF	007D2	:TPASKEYST		
												U.1191: .BLKB	0		
												007D3	:TPASKEYST		
37	5F	52	54	53	5F	45	54	41	56	49	50	007D3	:TPASKEYST		
												U.1193: .ASCII	\PRIVATE_STR_7\	:	
											FF	007E0	:TPASKEYST		
												U.1198: .BLKB	0		
												007E1	:TPASKEYST		

```

38 SF 52 54 53 SF 45 54 41 56 49 52 50 007E1 :TPASKEYST
      FF 007EE U.1200: .ASCII \PRIVATE_STR_8\
      007EF :TPASKEYSTO .BYTE -1
      007EF U.1205: .BLKB 0
39 SF 52 54 53 SF 45 54 41 56 49 52 50 007EF :TPASKEYST
      FF 007FC U.1207: .ASCII \PRIVATE_STR_9\
      007FD :TPASKEYSTO .BYTE -1
      007FD U.1212: .BLKB 0
30 31 SF 52 54 53 SF 45 54 41 56 49 52 50 007FD :TPASKEYST
      FF 0080B U.1214: .ASCII \PRIVATE_STR_10\
      FF 0080C :TPASKEYFILL .BYTE -1
      0080C U.1221: .BYTE -1

      .PSECT _LIB$STATES, NOWRT, SHR, PIC, 1
00000 SMG$A_STRING_STATES::
      .BLKB 0
00000 BEGIN_SCAN:
      .BLKB 0
99F8 00000 :TPATYPE
      U.2: .WORD -26120
0000* 00002 :TPASUBEXP
      U.4: .WORD <<U.3-U.4>-2>
00000000* 00004 :TPASACTION
      U.5: .LONG <<SMG$$NEXT_RECORD-U.5>-4>
0000* 00008 :TPASTARGET
      U.6: .WORD <<BEGIN_SCAN-U.6>-2>
9021 0000A :TPATYPE
      U.7: .WORD -28639
00000000* 0000C :TPASACTION
      U.8: .LONG <<SMG$$NEXT_RECORD-U.8>-4>
0000* 00010 :TPASTARGET
      U.9: .WORD <<BEGIN_SCAN-U.9>-2>
99F8 00012 :TPATYPE
      U.10: .WORD -26120
0000* 00014 :TPASUBEXP
      U.12: .WORD <<U.11-U.12>-2>
00000000* 00016 :TPASACTION
      U.13: .LONG <<SMG$$BLANKS_OFF-U.13>-4>
0000* 0001A :TPASTARGET
      U.14: .WORD <<BEGIN_SCAN-U.14>-2>
15F6 0001C :TPATYPE
      U.15: .WORD 5622
FFFF 0001E :TPASTARGET
      U.16: .WORD -1
00020 :END_OF_LINE
      U.3: .BLKB 0
11F7 00020 :TPATYPE
      U.17: .WORD 4599
FFFF 00022 :TPASTARGET
      U.18: .WORD -1
1021 00024 :TPATYPE
      U.19: .WORD 4129
FFFF 00026 :TPASTARGET

```

15F6	00028	U.20: .WORD	-1	:
		:TPASTYPE		:
FFFE	0002A	U.21: .WORD	5622	:
		:TPASTARGET		:
	0002C	U.22: .WORD	-2	:
		:CAPABILITY		:
99F8	0002C	U.11: .BLKB	0	:
		:TPASTYPE		:
0000*	0002E	U.23: .WORD	-26120	:
		:TPASSUBEXP		:
00000000*	00030	U.25: .WORD	<<U.24-U.25>-2>	:
		:TPASACTION		:
0000*	00034	U.26: .LONG	<<SMG\$\$BLANKS_OFF-U.26>-4>	:
		:TPASTARGET		:
1100	00036	U.28: .WORD	<<U.27-U.28>-2>	:
		:TPASTYPE		:
FFFE	00038	U.32: .WORD	4352	:
		:TPASTARGET		:
1101	0003A	U.33: .WORD	-2	:
		:TPASTYPE		:
FFFE	0003C	U.37: .WORD	4353	:
		:TPASTARGET		:
1102	0003E	U.38: .WORD	-2	:
		:TPASTYPE		:
FFFE	00040	U.42: .WORD	4354	:
		:TPASTARGET		:
1103	00042	U.43: .WORD	-2	:
		:TPASTYPE		:
0000*	00044	U.47: .WORD	4355	:
		:TPASTARGET		:
9104	00046	U.48: .WORD	<<BEGIN_SCAN-U.48>-2>	:
		:TPASTYPE		:
00000000*	00048	U.52: .WORD	-28412	:
		:TPASACTION		:
FFFE	0004C	U.53: .LONG	<<SMG\$\$MISSING_END-U.53>-4>	:
		:TPASTARGET		:
9105	0004E	U.54: .WORD	-2	:
		:TPASTYPE		:
00000000*	00050	U.58: .WORD	-28411	:
		:TPASACTION		:
FFFE	00054	U.59: .LONG	<<SMG\$\$MISSING_END-U.59>-4>	:
		:TPASTARGET		:
85F1	00056	U.60: .WORD	-2	:
		:TPASTYPE		:
00000000V	00058	U.61: .WORD	-31247	:
		:TPASACTION		:
	0005C	U.62: .LONG	<<NOT_STRING-U.62>-4>	:
		:STRING_NAME		:
7106	0005C	U.24: .BLKB	0	:
		:TPASTYPE		:
00000000*	0005E	U.67: .WORD	28934	:
		:TPASADDR		:
000001B9	00062	U.68: .LONG	<<SMG\$\$MASK_ADR-U.68>-4>	:
		:TPASMASK		:
FFFF	00066	U.69: .LONG	441	:
		:TPASTARGET		:
		U.70: .WORD	-1	:

Address	Mask	Offset	Field Name	Field Type	Value
7107	00068	U.74:	TPASTYPE	WORD	28935
00000000*	0006A	U.75:	TPAS\$ADDR	LONG	<<SMG\$\$MASK_ADR-U.75>-4>
000001BA	0006E	U.76:	TPAS\$MASK	LONG	442
FFFF	00072	U.77:	TPASTARGET	WORD	-1
7108	00074	U.81:	TPASTYPE	WORD	28936
00000000*	00076	U.82:	TPAS\$ADDR	LONG	<<SMG\$\$MASK_ADR-U.82>-4>
000001BB	0007A	U.83:	TPAS\$MASK	LONG	443
FFFF	0007E	U.84:	TPASTARGET	WORD	-1
7109	00080	U.88:	TPASTYPE	WORD	28937
00000000*	00082	U.89:	TPAS\$ADDR	LONG	<<SMG\$\$MASK_ADR-U.89>-4>
000001BC	00086	U.90:	TPAS\$MASK	LONG	444
FFFF	0008A	U.91:	TPASTARGET	WORD	-1
710A	0008C	U.95:	TPASTYPE	WORD	28938
00000000*	0008E	U.96:	TPAS\$ADDR	LONG	<<SMG\$\$MASK_ADR-U.96>-4>
000001BD	00092	U.97:	TPAS\$MASK	LONG	445
FFFF	00096	U.98:	TPASTARGET	WORD	-1
710B	00098	U.102:	TPASTYPE	WORD	28939
00000000*	0009A	U.103:	TPAS\$ADDR	LONG	<<SMG\$\$MASK_ADR-U.103>-4>
000001BE	0009E	U.104:	TPAS\$MASK	LONG	446
FFFF	000A2	U.105:	TPASTARGET	WORD	-1
710C	000A4	U.109:	TPASTYPE	WORD	28940
00000000*	000A6	U.110:	TPAS\$ADDR	LONG	<<SMG\$\$MASK_ADR-U.110>-4>
00000253	000AA	U.111:	TPAS\$MASK	LONG	595
FFFF	000AE	U.112:	TPASTARGET	WORD	-1
710D	000B0	U.116:	TPASTYPE	WORD	28941
00000000*	000B2	U.117:	TPAS\$ADDR	LONG	<<SMG\$\$MASK_ADR-U.117>-4>
000001BF	000B6	U.118:	TPAS\$MASK	LONG	447
FFFF	000BA	U.119:	TPASTARGET	WORD	-1
710E	000BC		TPASTYPE		

00000000*	000BE	U.123: .WORD	28942	:
		:TPASADDR		:
000001C0	000C2	U.124: .LONG	<<SMG\$MASK_ADR-U.124>-4>	:
		:TPASMASK		:
FFFF	000C6	U.125: .LONG	448	:
		:TPASTARGET		:
710F	000C8	U.126: .WORD	-1	:
		:TPASTYPE		:
00000000*	000CA	U.130: .WORD	28943	:
		:TPASADDR		:
000001C1	000CE	U.131: .LONG	<<SMG\$MASK_ADR-U.131>-4>	:
		:TPASMASK		:
FFFF	000D2	U.132: .LONG	449	:
		:TPASTARGET		:
7110	000D4	U.133: .WORD	-1	:
		:TPASTYPE		:
00000000*	000D6	U.137: .WORD	28944	:
		:TPASADDR		:
000001C2	000DA	U.138: .LONG	<<SMG\$MASK_ADR-U.138>-4>	:
		:TPASMASK		:
FFFF	000DE	U.139: .LONG	450	:
		:TPASTARGET		:
7111	000E0	U.140: .WORD	-1	:
		:TPASTYPE		:
00000000*	000E2	U.144: .WORD	28945	:
		:TPASADDR		:
00000249	000E6	U.145: .LONG	<<SMG\$MASK_ADR-U.145>-4>	:
		:TPASMASK		:
FFFF	000EA	U.146: .LONG	585	:
		:TPASTARGET		:
7112	000EC	U.147: .WORD	-1	:
		:TPASTYPE		:
00000000*	000EE	U.151: .WORD	28946	:
		:TPASADDR		:
000001C3	000F2	U.152: .LONG	<<SMG\$MASK_ADR-U.152>-4>	:
		:TPASMASK		:
FFFF	000F6	U.153: .LONG	451	:
		:TPASTARGET		:
7113	000F8	U.154: .WORD	-1	:
		:TPASTYPE		:
00000000*	000FA	U.158: .WORD	28947	:
		:TPASADDR		:
000001C4	000FE	U.159: .LONG	<<SMG\$MASK_ADR-U.159>-4>	:
		:TPASMASK		:
FFFF	00102	U.160: .LONG	452	:
		:TPASTARGET		:
7114	00104	U.161: .WORD	-1	:
		:TPASTYPE		:
00000000*	00106	U.165: .WORD	28948	:
		:TPASADDR		:
000001C5	0010A	U.166: .LONG	<<SMG\$MASK_ADR-U.166>-4>	:
		:TPASMASK		:
FFFF	0010E	U.167: .LONG	453	:
		:TPASTARGET		:
7115	00110	U.168: .WORD	-1	:
		:TPASTYPE		:
		U.172: .WORD	28949	:

00000000*	00112	:TPASADDR			
		U.173:	LONG	<<SMG\$MASK_ADR-U.173>-4>	:
0000024F	00116	:TPASMASK			
		U.174:	LONG	591	:
FFFF	0011A	:TPASTARGET			
		U.175:	WORD	-1	:
7116	0011C	:TPASTYPE			
		U.179:	WORD	28950	:
00000000*	0011E	:TPASADDR			
		U.180:	LONG	<<SMG\$MASK_ADR-U.180>-4>	:
00000247	00122	:TPASMASK			
		U.181:	LONG	583	:
FFFF	00126	:TPASTARGET			
		U.182:	WORD	-1	:
7117	00128	:TPASTYPE			
		U.186:	WORD	28951	:
00000000*	0012A	:TPASADDR			
		U.187:	LONG	<<SMG\$MASK_ADR-U.187>-4>	:
00000250	0012E	:TPASMASK			
		U.188:	LONG	592	:
FFFF	00132	:TPASTARGET			
		U.189:	WORD	-1	:
7118	00134	:TPASTYPE			
		U.193:	WORD	28952	:
00000000*	00136	:TPASADDR			
		U.194:	LONG	<<SMG\$MASK_ADR-U.194>-4>	:
000001C6	0013A	:TPASMASK			
		U.195:	LONG	454	:
FFFF	0013E	:TPASTARGET			
		U.196:	WORD	-1	:
7119	00140	:TPASTYPE			
		U.200:	WORD	28953	:
00000000*	00142	:TPASADDR			
		U.201:	LONG	<<SMG\$MASK_ADR-U.201>-4>	:
000001C7	00146	:TPASMASK			
		U.202:	LONG	455	:
FFFF	0014A	:TPASTARGET			
		U.203:	WORD	-1	:
711A	0014C	:TPASTYPE			
		U.207:	WORD	28954	:
00000000*	0014E	:TPASADDR			
		U.208:	LONG	<<SMG\$MASK_ADR-U.208>-4>	:
000001C8	00152	:TPASMASK			
		U.209:	LONG	456	:
FFFF	00156	:TPASTARGET			
		U.210:	WORD	-1	:
711B	00158	:TPASTYPE			
		U.214:	WORD	28955	:
00000000*	0015A	:TPASADDR			
		U.215:	LONG	<<SMG\$MASK_ADR-U.215>-4>	:
000001C9	0015E	:TPASMASK			
		U.216:	LONG	457	:
FFFF	00162	:TPASTARGET			
		U.217:	WORD	-1	:
711C	00164	:TPASTYPE			
		U.221:	WORD	28956	:
00000000*	00166	:TPASADDR			

000001CA	0016A	U.222: .LONG	<<SMG\$MASK_ADR-U.222>-4>	:
		:TP\$MASK		:
FFFF	0016E	U.223: .LONG	458	:
		:TP\$TARGET		:
711D	00170	U.224: .WORD	-1	:
		:TP\$TYPE		:
00000000*	00172	U.228: .WORD	28957	:
		:TP\$ADDR		:
000001CB	00176	U.229: .LONG	<<SMG\$MASK_ADR-U.229>-4>	:
		:TP\$MASK		:
FFFF	0017A	U.230: .LONG	459	:
		:TP\$TARGET		:
711E	0017C	U.231: .WORD	-1	:
		:TP\$TYPE		:
00000000*	0017E	U.235: .WORD	28958	:
		:TP\$ADDR		:
000001CC	00182	U.236: .LONG	<<SMG\$MASK_ADR-U.236>-4>	:
		:TP\$MASK		:
FFFF	00186	U.237: .LONG	460	:
		:TP\$TARGET		:
711F	00188	U.238: .WORD	-1	:
		:TP\$TYPE		:
00000000*	0018A	U.242: .WORD	28959	:
		:TP\$ADDR		:
000001CD	0018E	U.243: .LONG	<<SMG\$MASK_ADR-U.243>-4>	:
		:TP\$MASK		:
FFFF	00192	U.244: .LONG	461	:
		:TP\$TARGET		:
7120	00194	U.245: .WORD	-1	:
		:TP\$TYPE		:
00000000*	00196	U.249: .WORD	28960	:
		:TP\$ADDR		:
000001CE	0019A	U.250: .LONG	<<SMG\$MASK_ADR-U.250>-4>	:
		:TP\$MASK		:
FFFF	0019E	U.251: .LONG	462	:
		:TP\$TARGET		:
7121	001A0	U.252: .WORD	-1	:
		:TP\$TYPE		:
00000000*	001A2	U.256: .WORD	28961	:
		:TP\$ADDR		:
000001CF	001A6	U.257: .LONG	<<SMG\$MASK_ADR-U.257>-4>	:
		:TP\$MASK		:
FFFF	001AA	U.258: .LONG	463	:
		:TP\$TARGET		:
7122	001AC	U.259: .WORD	-1	:
		:TP\$TYPE		:
00000000*	001AE	U.263: .WORD	28962	:
		:TP\$ADDR		:
000001D0	001B2	U.264: .LONG	<<SMG\$MASK_ADR-U.264>-4>	:
		:TP\$MASK		:
FFFF	001B6	U.265: .LONG	464	:
		:TP\$TARGET		:
7123	001B8	U.266: .WORD	-1	:
		:TP\$TYPE		:
00000000*	001BA	U.270: .WORD	28963	:
		:TP\$ADDR		:
		U.271: .LONG	<<SMG\$MASK_ADR-U.271>-4>	:

000001D1	001BE	:TPASMASK		
		U.272: .LONG	465	:
FFFF	001C2	:TPASTARGET		:
		U.273: .WORD	-1	:
7124	001C4	:TPASTYPE		:
		U.277: .WORD	28964	:
00000000*	001C6	:TPASADDR		:
		U.278: .LONG	<<SMG\$\$MASK_ADR-U.278>-4>	:
000001D2	001CA	:TPASMASK		:
		U.279: .LONG	466	:
FFFF	001CE	:TPASTARGET		:
		U.280: .WORD	-1	:
7125	001D0	:TPASTYPE		:
		U.284: .WORD	28965	:
00000000*	001D2	:TPASADDR		:
		U.285: .LONG	<<SMG\$\$MASK_ADR-U.285>-4>	:
000001D3	001D6	:TPASMASK		:
		U.286: .LONG	467	:
FFFF	001DA	:TPASTARGET		:
		U.287: .WORD	-1	:
7126	001DC	:TPASTYPE		:
		U.291: .WORD	28966	:
00000000*	001DE	:TPASADDR		:
		U.292: .LONG	<<SMG\$\$MASK_ADR-U.292>-4>	:
000001D4	001E2	:TPASMASK		:
		U.293: .LONG	468	:
FFFF	001E6	:TPASTARGET		:
		U.294: .WORD	-1	:
7127	001E8	:TPASTYPE		:
		U.298: .WORD	28967	:
00000000*	001EA	:TPASADDR		:
		U.299: .LONG	<<SMG\$\$MASK_ADR-U.299>-4>	:
000001D5	001EE	:TPASMASK		:
		U.300: .LONG	469	:
FFFF	001F2	:TPASTARGET		:
		U.301: .WORD	-1	:
7128	001F4	:TPASTYPE		:
		U.305: .WORD	28968	:
00000000*	001F6	:TPASADDR		:
		U.306: .LONG	<<SMG\$\$MASK_ADR-U.306>-4>	:
000001D6	001FA	:TPASMASK		:
		U.307: .LONG	470	:
FFFF	001FE	:TPASTARGET		:
		U.308: .WORD	-1	:
7129	00200	:TPASTYPE		:
		U.312: .WORD	28969	:
00000000*	00202	:TPASADDR		:
		U.313: .LONG	<<SMG\$\$MASK_ADR-U.313>-4>	:
000001D7	00206	:TPASMASK		:
		U.314: .LONG	471	:
FFFF	0020A	:TPASTARGET		:
		U.315: .WORD	-1	:
712A	0020C	:TPASTYPE		:
		U.319: .WORD	28970	:
00000000*	0020E	:TPASADDR		:
		U.320: .LONG	<<SMG\$\$MASK_ADR-U.320>-4>	:
000001D8	00212	:TPASMASK		:

		U.321: .LONG	472	:
FFFF	00216	:TPASTARGET		:
		U.322: .WORD	-1	:
712B	00218	:TPASTYPE		:
		U.326: .WORD	28971	:
00000000*	0021A	:TPASADDR		:
		U.327: .LONG	<<SMG\$MASK_ADR-U.327>-4>	:
000001D9	0021E	:TPASMASK		:
		U.328: .LONG	473	:
FFFF	00222	:TPASTARGET		:
		U.329: .WORD	-1	:
712C	00224	:TPASTYPE		:
		U.333: .WORD	28972	:
00000000*	00226	:TPASADDR		:
		U.334: .LONG	<<SMG\$MASK_ADR-U.334>-4>	:
000001DA	0022A	:TPASMASK		:
		U.335: .LONG	474	:
FFFF	0022E	:TPASTARGET		:
		U.336: .WORD	-1	:
712D	00230	:TPASTYPE		:
		U.340: .WORD	28973	:
00000000*	00232	:TPASADDR		:
		U.341: .LONG	<<SMG\$MASK_ADR-U.341>-4>	:
000001DB	00236	:TPASMASK		:
		U.342: .LONG	475	:
FFFF	0023A	:TPASTARGET		:
		U.343: .WORD	-1	:
712E	0023C	:TPASTYPE		:
		U.347: .WORD	28974	:
00000000*	0023E	:TPASADDR		:
		U.348: .LONG	<<SMG\$MASK_ADR-U.348>-4>	:
0000024A	00242	:TPASMASK		:
		U.349: .LONG	586	:
FFFF	00246	:TPASTARGET		:
		U.350: .WORD	-1	:
712F	00248	:TPASTYPE		:
		U.354: .WORD	28975	:
00000000*	0024A	:TPASADDR		:
		U.355: .LONG	<<SMG\$MASK_ADR-U.355>-4>	:
000001DC	0024E	:TPASMASK		:
		U.356: .LONG	476	:
FFFF	00252	:TPASTARGET		:
		U.357: .WORD	-1	:
7130	00254	:TPASTYPE		:
		U.361: .WORD	28976	:
00000000*	00256	:TPASADDR		:
		U.362: .LONG	<<SMG\$MASK_ADR-U.362>-4>	:
000001DD	0025A	:TPASMASK		:
		U.363: .LONG	477	:
FFFF	0025E	:TPASTARGET		:
		U.364: .WORD	-1	:
7131	00260	:TPASTYPE		:
		U.368: .WORD	28977	:
00000000*	00262	:TPASADDR		:
		U.369: .LONG	<<SMG\$MASK_ADR-U.369>-4>	:
0000024C	00266	:TPASMASK		:
		U.370: .LONG	588	:

FFFF	0026A	:TPASTARGET		
		U.371: .WORD	-1	:
7132	0026C	:TPASTYPE		:
		U.375: .WORD	28978	:
00000000*	0026E	:TPASADDR		:
		U.376: .LONG	<<SMG\$MASK_ADR-U.376>-4>	:
00000251	00272	:TPASMASK		:
		U.377: .LONG	593	:
FFFF	00276	:TPASTARGET		:
		U.378: .WORD	-1	:
7133	00278	:TPASTYPE		:
		U.382: .WORD	28979	:
00000000*	0027A	:TPASADDR		:
		U.383: .LONG	<<SMG\$MASK_ADR-U.383>-4>	:
000001DE	0027E	:TPASMASK		:
		U.384: .LONG	478	:
FFFF	00282	:TPASTARGET		:
		U.385: .WORD	-1	:
7134	00284	:TPASTYPE		:
		U.389: .WORD	28980	:
00000000*	00286	:TPASADDR		:
		U.390: .LONG	<<SMG\$MASK_ADR-U.390>-4>	:
000001DF	0028A	:TPASMASK		:
		U.391: .LONG	479	:
FFFF	0028E	:TPASTARGET		:
		U.392: .WORD	-1	:
7135	00290	:TPASTYPE		:
		U.396: .WORD	28981	:
00000000*	00292	:TPASADDR		:
		U.397: .LONG	<<SMG\$MASK_ADR-U.397>-4>	:
000001E0	00296	:TPASMASK		:
		U.398: .LONG	480	:
FFFF	0029A	:TPASTARGET		:
		U.399: .WORD	-1	:
7136	0029C	:TPASTYPE		:
		U.403: .WORD	28982	:
00000000*	0029E	:TPASADDR		:
		U.404: .LONG	<<SMG\$MASK_ADR-U.404>-4>	:
000001E1	002A2	:TPASMASK		:
		U.405: .LONG	481	:
FFFF	002A6	:TPASTARGET		:
		U.406: .WORD	-1	:
7137	002A8	:TPASTYPE		:
		U.410: .WORD	28983	:
00000000*	002AA	:TPASADDR		:
		U.411: .LONG	<<SMG\$MASK_ADR-U.411>-4>	:
000001E2	002AE	:TPASMASK		:
		U.412: .LONG	482	:
FFFF	002B2	:TPASTARGET		:
		U.413: .WORD	-1	:
7138	002B4	:TPASTYPE		:
		U.417: .WORD	28984	:
00000000*	002B6	:TPASADDR		:
		U.418: .LONG	<<SMG\$MASK_ADR-U.418>-4>	:
000001E3	002BA	:TPASMASK		:
		U.419: .LONG	483	:
FFFF	002BE	:TPASTARGET		:

		U.420:	WORD	-1	:
7139	002C0	:TPASTYPE			:
		U.424:	WORD	28985	:
00000000*	002C2	:TPASADDR			:
		U.425:	LONG	<<SMG\$MASK_ADR-U.425>-4>	:
000001E4	002C6	:TPASMASK			:
		U.426:	LONG	484	:
FFFF	002CA	:TPASTARGET			:
		U.427:	WORD	-1	:
713A	002CC	:TPASTYPE			:
		U.431:	WORD	28986	:
00000000*	002CE	:TPASADDR			:
		U.432:	LONG	<<SMG\$MASK_ADR-U.432>-4>	:
000001E5	002D2	:TPASMASK			:
		U.433:	LONG	485	:
FFFF	002D6	:TPASTARGET			:
		U.434:	WORD	-1	:
713B	002D8	:TPASTYPE			:
		U.438:	WORD	28987	:
00000000*	002DA	:TPASADDR			:
		U.439:	LONG	<<SMG\$MASK_ADR-U.439>-4>	:
000001E6	002DE	:TPASMASK			:
		U.440:	LONG	486	:
FFFF	002E2	:TPASTARGET			:
		U.441:	WORD	-1	:
713C	002E4	:TPASTYPE			:
		U.445:	WORD	28988	:
00000000*	002E6	:TPASADDR			:
		U.446:	LONG	<<SMG\$MASK_ADR-U.446>-4>	:
000001E7	002EA	:TPASMASK			:
		U.447:	LONG	487	:
FFFF	002EE	:TPASTARGET			:
		U.448:	WORD	-1	:
713D	002F0	:TPASTYPE			:
		U.452:	WORD	28989	:
00000000*	002F2	:TPASADDR			:
		U.453:	LONG	<<SMG\$MASK_ADR-U.453>-4>	:
000001E8	002F6	:TPASMASK			:
		U.454:	LONG	488	:
FFFF	002FA	:TPASTARGET			:
		U.455:	WORD	-1	:
713E	002FC	:TPASTYPE			:
		U.459:	WORD	28990	:
00000000*	002FE	:TPASADDR			:
		U.460:	LONG	<<SMG\$MASK_ADR-U.460>-4>	:
000001E9	00302	:TPASMASK			:
		U.461:	LONG	489	:
FFFF	00306	:TPASTARGET			:
		U.462:	WORD	-1	:
713F	00308	:TPASTYPE			:
		U.466:	WORD	28991	:
00000000*	0030A	:TPASADDR			:
		U.467:	LONG	<<SMG\$MASK_ADR-U.467>-4>	:
000001EA	0030E	:TPASMASK			:
		U.468:	LONG	490	:
FFFF	00312	:TPASTARGET			:
		U.469:	WORD	-1	:

7140	00314	:TPASTYPE			
		U.473:	WORD	28992	:
00000000*	00316	:TPASADDR			:
		U.474:	LONG	<<SMG\$MASK_ADR-U.474>-4>	:
000001EB	0031A	:TPASMASK			:
		U.475:	LONG	491	:
FFFF	0031E	:TPASTARGET			:
		U.476:	WORD	-1	:
7141	00320	:TPASTYPE			:
		U.480:	WORD	28993	:
00000000*	00322	:TPASADDR			:
		U.481:	LONG	<<SMG\$MASK_ADR-U.481>-4>	:
000001EC	00326	:TPASMASK			:
		U.482:	LONG	492	:
FFFF	0032A	:TPASTARGET			:
		U.483:	WORD	-1	:
7142	0032C	:TPASTYPE			:
		U.487:	WORD	28994	:
00000000*	0032E	:TPASADDR			:
		U.488:	LONG	<<SMG\$MASK_ADR-U.488>-4>	:
000001ED	00332	:TPASMASK			:
		U.489:	LONG	493	:
FFFF	00336	:TPASTARGET			:
		U.490:	WORD	-1	:
7143	00338	:TPASTYPE			:
		U.494:	WORD	28995	:
000000C0*	0033A	:TPASADDR			:
		U.495:	LONG	<<SMG\$MASK_ADR-U.495>-4>	:
000001EE	0033E	:TPASMASK			:
		U.496:	LONG	494	:
FFFF	00342	:TPASTARGET			:
		U.497:	WORD	-1	:
7144	00344	:TPASTYPE			:
		U.501:	WORD	28996	:
00000000*	00346	:TPASADDR			:
		U.502:	LONG	<<SMG\$MASK_ADR-U.502>-4>	:
000001EF	0034A	:TPASMASK			:
		U.503:	LONG	495	:
FFFF	0034E	:TPASTARGET			:
		U.504:	WORD	-1	:
7145	00350	:TPASTYPE			:
		U.508:	WORD	28997	:
00000000*	00352	:TPASADDR			:
		U.509:	LONG	<<SMG\$MASK_ADR-U.509>-4>	:
000001F0	00356	:TPASMASK			:
		U.510:	LONG	496	:
FFFF	0035A	:TPASTARGET			:
		U.511:	WORD	-1	:
7146	0035C	:TPASTYPE			:
		U.515:	WORD	28998	:
00000000*	0035E	:TPASADDR			:
		U.516:	LONG	<<SMG\$MASK_ADR-U.516>-4>	:
000001F1	00362	:TPASMASK			:
		U.517:	LONG	497	:
FFFF	00366	:TPASTARGET			:
		U.518:	WORD	-1	:
7147	00368	:TPASTYPE			:

00000000*	0036A	U.522: .WORD	28999	:
		:TPAS\$ADDR		:
000001F2	0036E	U.523: .LONG	<<SMG\$\$MASK_ADR-U.523>-4>	:
		:TPAS\$MASK		:
FFFF	00372	U.524: .LONG	498	:
		:TPAS\$TARGET		:
7148	00374	U.525: .WORD	-1	:
		:TPAS\$TYPE		:
00000000*	00376	U.529: .WORD	29000	:
		:TPAS\$ADDR		:
000001F3	0037A	U.530: .LONG	<<SMG\$\$MASK_ADR-U.530>-4>	:
		:TPAS\$MASK		:
FFFF	0037E	U.531: .LONG	499	:
		:TPAS\$TARGET		:
7149	00380	U.532: .WORD	-1	:
		:TPAS\$TYPE		:
00000000*	00382	U.536: .WORD	29001	:
		:TPAS\$ADDR		:
000001F4	00386	U.537: .LONG	<<SMG\$\$MASK_ADR-U.537>-4>	:
		:TPAS\$MASK		:
FFFF	0038A	U.538: .LONG	500	:
		:TPAS\$TARGET		:
714A	0038C	U.539: .WORD	-1	:
		:TPAS\$TYPE		:
00000000*	0038E	U.543: .WORD	29002	:
		:TPAS\$ADDR		:
000001F5	00392	U.544: .LONG	<<SMG\$\$MASK_ADR-U.544>-4>	:
		:TPAS\$MASK		:
FFFF	00396	U.545: .LONG	501	:
		:TPAS\$TARGET		:
714B	00398	U.546: .WORD	-1	:
		:TPAS\$TYPE		:
00000000*	0039A	U.550: .WORD	29003	:
		:TPAS\$ADDR		:
000001F6	0039E	U.551: .LONG	<<SMG\$\$MASK_ADR-U.551>-4>	:
		:TPAS\$MASK		:
FFFF	003A2	U.552: .LONG	502	:
		:TPAS\$TARGET		:
714C	003A4	U.553: .WORD	-1	:
		:TPAS\$TYPE		:
00000000*	003A6	U.557: .WORD	29004	:
		:TPAS\$ADDR		:
000001F7	003AA	U.558: .LONG	<<SMG\$\$MASK_ADR-U.558>-4>	:
		:TPAS\$MASK		:
FFFF	003AE	U.559: .LONG	503	:
		:TPAS\$TARGET		:
714D	003B0	U.560: .WORD	-1	:
		:TPAS\$TYPE		:
00000000*	003B2	U.564: .WORD	29005	:
		:TPAS\$ADDR		:
000001F8	003B6	U.565: .LONG	<<SMG\$\$MASK_ADR-U.565>-4>	:
		:TPAS\$MASK		:
FFFF	003BA	U.566: .LONG	504	:
		:TPAS\$TARGET		:
714E	003BC	U.567: .WORD	-1	:
		:TPAS\$TYPE		:
		U.571: .WORD	29006	:

00000000*	003BE	:TPAS\$ADDR			
		U.572:	LONG	<<SMG\$MASK_ADR-U.572>-4>	:
000001F9	003C2	:TPAS\$MASK			:
		U.573:	LONG	505	:
FFFF	003C6	:TPAS\$TARGET			:
		U.574:	WORD	-1	:
714F	003C8	:TPAS\$TYPE			:
		U.578:	WORD	29007	:
00000000*	003CA	:TPAS\$ADDR			:
		U.579:	LONG	<<SMG\$MASK_ADR-U.579>-4>	:
000001FA	003CE	:TPAS\$MASK			:
		U.580:	LONG	506	:
FFFF	003D2	:TPAS\$TARGET			:
		U.581:	WORD	-1	:
7150	003D4	:TPAS\$TYPE			:
		U.585:	WORD	29008	:
00000000*	003D6	:TPAS\$ADDR			:
		U.586:	LONG	<<SMG\$MASK_ADR-U.586>-4>	:
000001FB	003DA	:TPAS\$MASK			:
		U.587:	LONG	507	:
FFFF	003DE	:TPAS\$TARGET			:
		U.588:	WORD	-1	:
7151	003E0	:TPAS\$TYPE			:
		U.592:	WORD	29009	:
00000000*	003E2	:TPAS\$ADDR			:
		U.593:	LONG	<<SMG\$MASK_ADR-U.593>-4>	:
000001FC	003E6	:TPAS\$MASK			:
		U.594:	LONG	508	:
FFFF	003EA	:TPAS\$TARGET			:
		U.595:	WORD	-1	:
7152	003EC	:TPAS\$TYPE			:
		U.599:	WORD	29010	:
00000000*	003EE	:TPAS\$ADDR			:
		U.600:	LONG	<<SMG\$MASK_ADR-U.600>-4>	:
000001FD	003F2	:TPAS\$MASK			:
		U.601:	LONG	509	:
FFFF	003F6	:TPAS\$TARGET			:
		U.602:	WORD	-1	:
7153	003F8	:TPAS\$TYPE			:
		U.606:	WORD	29011	:
00000000*	003FA	:TPAS\$ADDR			:
		U.607:	LONG	<<SMG\$MASK_ADR-U.607>-4>	:
000001FE	003FE	:TPAS\$MASK			:
		U.608:	LONG	510	:
FFFF	00402	:TPAS\$TARGET			:
		U.609:	WORD	-1	:
7154	00404	:TPAS\$TYPE			:
		U.613:	WORD	29012	:
00000000*	00406	:TPAS\$ADDR			:
		U.614:	LONG	<<SMG\$MASK_ADR-U.614>-4>	:
000001FF	0040A	:TPAS\$MASK			:
		U.615:	LONG	511	:
FFFF	0040E	:TPAS\$TARGET			:
		U.616:	WORD	-1	:
7155	00410	:TPAS\$TYPE			:
		U.620:	WORD	29013	:
00000000*	00412	:TPAS\$ADDR			:

00000200	00416	U.621: .LONG	<<SMG\$MASK_ADR-U.621>-4>	:
		:TP\$MASK		:
FFFF	0041A	U.622: .LONG	512	:
		:TP\$TARGET		:
7156	0041C	U.623: .WORD	-1	:
		:TP\$TYPE		:
00000000*	0041E	U.627: .WORD	29014	:
		:TP\$ADDR		:
00000201	00422	U.628: .LONG	<<SMG\$MASK_ADR-U.628>-4>	:
		:TP\$MASK		:
FFFF	00426	U.629: .LONG	513	:
		:TP\$TARGET		:
		U.630: .WORD	-1	:
7157	00428	:TP\$TYPE		:
		U.634: .WORD	29015	:
00000000*	0042A	:TP\$ADDR		:
		U.635: .LONG	<<SMG\$MASK_ADR-U.635>-4>	:
00000202	0042E	:TP\$MASK		:
		U.636: .LONG	514	:
FFFF	00432	:TP\$TARGET		:
		U.637: .WORD	-1	:
7158	00434	:TP\$TYPE		:
		U.641: .WORD	29016	:
00000000*	00436	:TP\$ADDR		:
		U.642: .LONG	<<SMG\$MASK_ADR-U.642>-4>	:
00000203	0043A	:TP\$MASK		:
		U.643: .LONG	515	:
FFFF	0043E	:TP\$TARGET		:
		U.644: .WORD	-1	:
7159	00440	:TP\$TYPE		:
		U.648: .WORD	29017	:
00000000*	00442	:TP\$ADDR		:
		U.649: .LONG	<<SMG\$MASK_ADR-U.649>-4>	:
00000204	00446	:TP\$MASK		:
		U.650: .LONG	516	:
FFFF	0044A	:TP\$TARGET		:
		U.651: .WORD	-1	:
715A	0044C	:TP\$TYPE		:
		U.655: .WORD	29018	:
00000000*	0044E	:TP\$ADDR		:
		U.656: .LONG	<<SMG\$MASK_ADR-U.656>-4>	:
00000205	00452	:TP\$MASK		:
		U.657: .LONG	517	:
FFFF	00456	:TP\$TARGET		:
		U.658: .WORD	-1	:
715B	00458	:TP\$TYPE		:
		U.662: .WORD	29019	:
00000000*	0045A	:TP\$ADDR		:
		U.663: .LONG	<<SMG\$MASK_ADR-U.663>-4>	:
00000206	0045E	:TP\$MASK		:
		U.664: .LONG	518	:
FFFF	00462	:TP\$TARGET		:
		U.665: .WORD	-1	:
715C	00464	:TP\$TYPE		:
		U.669: .WORD	29020	:
00000000*	00466	:TP\$ADDR		:
		U.670: .LONG	<<SMG\$MASK_ADR-U.670>-4>	:

00000207	0046A	:TPASMASK		
		U.671: .LONG	519	:
FFFF	0046E	:TPASTARGET		:
		U.672: .WORD	-1	:
715D	00470	:TPASTYPE		:
		U.676: .WORD	29021	:
00000000*	00472	:TPASADDR		:
		U.677: .LONG	<<SMG\$SMASK_ADR-U.677>-4>	:
00000208	00476	:TPASMASK		:
		U.678: .LONG	520	:
FFFF	0047A	:TPASTARGET		:
		U.679: .WORD	-1	:
715E	0047C	:TPASTYPE		:
		U.683: .WORD	29022	:
00000000*	0047E	:TPASADDR		:
		U.684: .LONG	<<SMG\$SMASK_ADR-U.684>-4>	:
00000209	00482	:TPASMASK		:
		U.685: .LONG	521	:
FFFF	00486	:TPASTARGET		:
		U.686: .WORD	-1	:
715F	00488	:TPASTYPE		:
		U.690: .WORD	29023	:
00000000*	0048A	:TPASADDR		:
		U.691: .LONG	<<SMG\$SMASK_ADR-U.691>-4>	:
0000020A	0048E	:TPASMASK		:
		U.692: .LONG	522	:
FFFF	00492	:TPASTARGET		:
		U.693: .WORD	-1	:
7160	00494	:TPASTYPE		:
		U.697: .WORD	29024	:
00000000*	00496	:TPASADDR		:
		U.698: .LONG	<<SMG\$SMASK_ADR-U.698>-4>	:
0000020B	0049A	:TPASMASK		:
		U.699: .LONG	523	:
FFFF	0049E	:TPASTARGET		:
		U.700: .WORD	-1	:
7161	004A0	:TPASTYPE		:
		U.704: .WORD	29025	:
00000000*	004A2	:TPASADDR		:
		U.705: .LONG	<<SMG\$SMASK_ADR-U.705>-4>	:
0000020C	004A6	:TPASMASK		:
		U.706: .LONG	524	:
FFFF	004AA	:TPASTARGET		:
		U.707: .WORD	-1	:
7162	004AC	:TPASTYPE		:
		U.711: .WORD	29026	:
00000000*	004AE	:TPASADDR		:
		U.712: .LONG	<<SMG\$SMASK_ADR-U.712>-4>	:
0000020D	004B2	:TPASMASK		:
		U.713: .LONG	525	:
FFFF	004B6	:TPASTARGET		:
		U.714: .WORD	-1	:
7163	004B8	:TPASTYPE		:
		U.718: .WORD	29027	:
00000000*	004BA	:TPASADDR		:
		U.719: .LONG	<<SMG\$SMASK_ADR-U.719>-4>	:
0000020E	004BE	:TPASMASK		:

		U.720:	.LONG	526	:
FFFF	004C2	:TPAS	TARGET		:
		U.721:	.WORD	-1	:
7164	004C4	:TPAS	TYPE		:
		U.725:	.WORD	29028	:
00000000*	004C6	:TPAS	ADDR		:
		U.726:	.LONG	<<SMG\$MASK_ADR-U.726>-4>	:
0000020F	004CA	:TPAS	MASK		:
		U.727:	.LONG	527	:
FFFF	004CE	:TPAS	TARGET		:
		U.728:	.WORD	-1	:
7165	004D0	:TPAS	TYPE		:
		U.732:	.WORD	29029	:
00000000*	004D2	:TPAS	ADDR		:
		U.733:	.LONG	<<SMG\$MASK_ADR-U.733>-4>	:
00000210	004D6	:TPAS	MASK		:
		U.734:	.LONG	528	:
FFFF	004DA	:TPAS	TARGET		:
		U.735:	.WORD	-1	:
7166	004DC	:TPAS	TYPE		:
		U.739:	.WORD	29030	:
00000000*	004DE	:TPAS	ADDR		:
		U.740:	.LONG	<<SMG\$MASK_ADR-U.740>-4>	:
00000211	004E2	:TPAS	MASK		:
		U.741:	.LONG	529	:
FFFF	004E6	:TPAS	TARGET		:
		U.742:	.WORD	-1	:
7167	004E8	:TPAS	TYPE		:
		U.746:	.WORD	29031	:
00000000*	004EA	:TPAS	ADDR		:
		U.747:	.LONG	<<SMG\$MASK_ADR-U.747>-4>	:
00000212	004EE	:TPAS	MASK		:
		U.748:	.LONG	530	:
FFFF	004F2	:TPAS	TARGET		:
		U.749:	.WORD	-1	:
7168	004F4	:TPAS	TYPE		:
		U.753:	.WORD	29032	:
00000000*	004F6	:TPAS	ADDR		:
		U.754:	.LONG	<<SMG\$MASK_ADR-U.754>-4>	:
00000213	004FA	:TPAS	MASK		:
		U.755:	.LONG	531	:
FFFF	004FE	:TPAS	TARGET		:
		U.756:	.WORD	-1	:
7169	00500	:TPAS	TYPE		:
		U.760:	.WORD	29033	:
00000000*	00502	:TPAS	ADDR		:
		U.761:	.LONG	<<SMG\$MASK_ADR-U.761>-4>	:
00000214	00506	:TPAS	MASK		:
		U.762:	.LONG	532	:
FFFF	0050A	:TPAS	TARGET		:
		U.763:	.WORD	-1	:
716A	0050C	:TPAS	TYPE		:
		U.767:	.WORD	29034	:
00000000*	0050E	:TPAS	ADDR		:
		U.768:	.LONG	<<SMG\$MASK_ADR-U.768>-4>	:
00000215	00512	:TPAS	MASK		:
		U.769:	.LONG	533	:

FFFF	00516	:TPASTARGET			
		U.770: .WORD	-1	:	
716B	00518	:TPASTYPE		:	
		U.774: .WORD	29035	:	
00000000*	0051A	:TPASADDR		:	
		U.775: .LONG	<<SMG\$MASK_ADR-U.775>-4>	:	
00000216	0051E	:TPASMASK		:	
		U.776: .LONG	534	:	
FFFF	00522	:TPASTARGET		:	
		U.777: .WORD	-1	:	
716C	00524	:TPASTYPE		:	
		U.781: .WORD	29036	:	
00000000*	00526	:TPASADDR		:	
		U.782: .LONG	<<SMG\$MASK_ADR-U.782>-4>	:	
00000217	0052A	:TPASMASK		:	
		U.783: .LONG	535	:	
FFFF	0052E	:TPASTARGET		:	
		U.784: .WORD	-1	:	
716D	00530	:TPASTYPE		:	
		U.788: .WORD	29037	:	
00000000*	00532	:TPASADDR		:	
		U.789: .LONG	<<SMG\$MASK_ADR-U.789>-4>	:	
00000218	00536	:TPASMASK		:	
		U.790: .LONG	536	:	
FFFF	0053A	:TPASTARGET		:	
		U.791: .WORD	-1	:	
716E	0053C	:TPASTYPE		:	
		U.795: .WORD	29038	:	
00000000*	0053E	:TPASADDR		:	
		U.796: .LONG	<<SMG\$MASK_ADR-U.796>-4>	:	
00000219	00542	:TPASMASK		:	
		U.797: .LONG	537	:	
FFFF	00546	:TPASTARGET		:	
		U.798: .WORD	-1	:	
716F	00548	:TPASTYPE		:	
		U.802: .WORD	29039	:	
00000000*	0054A	:TPASADDR		:	
		U.803: .LONG	<<SMG\$MASK_ADR-U.803>-4>	:	
0000021A	0054E	:TPASMASK		:	
		U.804: .LONG	538	:	
FFFF	00552	:TPASTARGET		:	
		U.805: .WORD	-1	:	
7170	00554	:TPASTYPE		:	
		U.809: .WORD	29040	:	
00000000*	00556	:TPASADDR		:	
		U.810: .LONG	<<SMG\$MASK_ADR-U.810>-4>	:	
0000021B	0055A	:TPASMASK		:	
		U.811: .LONG	539	:	
FFFF	0055E	:TPASTARGET		:	
		U.812: .WORD	-1	:	
7171	00560	:TPASTYPE		:	
		U.816: .WORD	29041	:	
00000000*	00562	:TPASADDR		:	
		U.817: .LONG	<<SMG\$MASK_ADR-U.817>-4>	:	
0000021C	00566	:TPASMASK		:	
		U.818: .LONG	540	:	
FFFF	0056A	:TPASTARGET		:	

7172	0056C	U.819: .WORD	-1	:
		:TPA\$TYPE		:
00000000*	0056E	U.823: .WORD	29042	:
		:TPA\$ADDR		:
0000021D	00572	U.824: .LONG	<<SMG\$SMASK_ADR-U.824>-4>	:
		:TPA\$MASK		:
FFFF	00576	U.825: .LONG	541	:
		:TPA\$TARGET		:
7173	00578	U.826: .WORD	-1	:
		:TPA\$TYPE		:
00000000*	0057A	U.830: .WORD	29043	:
		:TPA\$ADDR		:
0000021E	0057E	U.831: .LONG	<<SMG\$SMASK_ADR-U.831>-4>	:
		:TPA\$MASK		:
FFFF	00582	U.832: .LONG	542	:
		:TPA\$TARGET		:
7174	00584	U.833: .WORD	-1	:
		:TPA\$TYPE		:
00000000*	00586	U.837: .WORD	29044	:
		:TPA\$ADDR		:
0000021F	0058A	U.838: .LONG	<<SMG\$SMASK_ADR-U.838>-4>	:
		:TPA\$MASK		:
FFFF	0058E	U.839: .LONG	543	:
		:TPA\$TARGET		:
7175	00590	U.840: .WORD	-1	:
		:TPA\$TYPE		:
00000000*	00592	U.844: .WORD	29045	:
		:TPA\$ADDR		:
00000220	00596	U.845: .LONG	<<SMG\$SMASK_ADR-U.845>-4>	:
		:TPA\$MASK		:
FFFF	0059A	U.846: .LONG	544	:
		:TPA\$TARGET		:
7176	0059C	U.847: .WORD	-1	:
		:TPA\$TYPE		:
00000000*	0059E	U.851: .WORD	29046	:
		:TPA\$ADDR		:
00000221	005A2	U.852: .LONG	<<SMG\$SMASK_ADR-U.852>-4>	:
		:TPA\$MASK		:
FFFF	005A6	U.853: .LONG	545	:
		:TPA\$TARGET		:
7177	005A8	U.854: .WORD	-1	:
		:TPA\$TYPE		:
00000000*	005AA	U.858: .WORD	29047	:
		:TPA\$ADDR		:
00000222	005AE	U.859: .LONG	<<SMG\$SMASK_ADR-U.859>-4>	:
		:TPA\$MASK		:
FFFF	005B2	U.860: .LONG	546	:
		:TPA\$TARGET		:
7178	005B4	U.861: .WORD	-1	:
		:TPA\$TYPE		:
00000000*	005B6	U.865: .WORD	29048	:
		:TPA\$ADDR		:
00000223	005BA	U.866: .LONG	<<SMG\$SMASK_ADR-U.866>-4>	:
		:TPA\$MASK		:
FFFF	005BE	U.867: .LONG	547	:
		:TPA\$TARGET		:
		U.868: .WORD	-1	:

7179	005C0	:TPATYPE	
		U.872: .WORD	29049
00000000*	005C2	:TPASADDR	
		U.873: .LONG	<<SMG\$MASK_ADR-U.873>-4>
00000224	005C6	:TPASMASK	
		U.874: .LONG	548
FFFF	005CA	:TPATARGET	
		U.875: .WORD	-1
717A	005CC	:TPATYPE	
		U.879: .WORD	29050
00000000*	005CE	:TPASADDR	
		U.880: .LONG	<<SMG\$MASK_ADR-U.880>-4>
00000225	005D2	:TPASMASK	
		U.881: .LONG	549
FFFF	005D6	:TPATARGET	
		U.882: .WORD	-1
717B	005D8	:TPATYPE	
		U.886: .WORD	29051
00000000*	005DA	:TPASADDR	
		U.887: .LONG	<<SMG\$MASK_ADR-U.887>-4>
00000226	005DE	:TPASMASK	
		U.888: .LONG	550
FFFF	005E2	:TPATARGET	
		U.889: .WORD	-1
717C	005E4	:TPATYPE	
		U.893: .WORD	29052
00000000*	005E6	:TPASADDR	
		U.894: .LONG	<<SMG\$MASK_ADR-U.894>-4>
00000227	005EA	:TPASMASK	
		U.895: .LONG	551
FFFF	005EE	:TPATARGET	
		U.896: .WORD	-1
717D	005F0	:TPATYPE	
		U.900: .WORD	29053
00000000*	005F2	:TPASADDR	
		U.901: .LONG	<<SMG\$MASK_ADR-U.901>-4>
0000024B	005F6	:TPASMASK	
		U.902: .LONG	587
FFFF	005FA	:TPATARGET	
		U.903: .WORD	-1
717E	005FC	:TPATYPE	
		U.907: .WORD	29054
00000000*	005FE	:TPASADDR	
		U.908: .LONG	<<SMG\$MASK_ADR-U.908>-4>
00000228	00602	:TPASMASK	
		U.909: .LONG	552
FFFF	00606	:TPATARGET	
		U.910: .WORD	-1
717F	00608	:TPATYPE	
		U.914: .WORD	29055
00000000*	0060A	:TPASADDR	
		U.915: .LONG	<<SMG\$MASK_ADR-U.915>-4>
00000229	0060E	:TPASMASK	
		U.916: .LONG	553
FFFF	00612	:TPATARGET	
		U.917: .WORD	-1
7180	00614	:TPATYPE	

00000000*	00616	U.921: .WORD	29056	:
		:TPASADDR		:
0000022A	0061A	U.922: .LONG	<<SMG\$MASK_ADR-U.922>-4>	:
		:TPASMASK		:
FFFF	0061E	U.923: .LONG	554	:
		:TPASTARGET		:
7181	00620	U.924: .WORD	-1	:
		:TPASTYPE		:
00000000*	00622	U.928: .WORD	29057	:
		:TPASADDR		:
0000022B	00626	U.929: .LONG	<<SMG\$MASK_ADR-U.929>-4>	:
		:TPASMASK		:
FFFF	0062A	U.930: .LONG	555	:
		:TPASTARGET		:
7182	0062C	U.931: .WORD	-1	:
		:TPASTYPE		:
00000000*	0062E	U.935: .WORD	29058	:
		:TPASADDR		:
0000022C	00632	U.936: .LONG	<<SMG\$MASK_ADR-U.936>-4>	:
		:TPASMASK		:
FFFF	00636	U.937: .LONG	556	:
		:TPASTARGET		:
7183	00638	U.938: .WORD	-1	:
		:TPASTYPE		:
00000000*	0063A	U.942: .WORD	29059	:
		:TPASADDR		:
0000022D	0063E	U.943: .LONG	<<SMG\$MASK_ADR-U.943>-4>	:
		:TPASMASK		:
FFFF	00642	U.944: .LONG	557	:
		:TPASTARGET		:
7184	00644	U.945: .WORD	-1	:
		:TPASTYPE		:
00000000*	00646	U.949: .WORD	29060	:
		:TPASADDR		:
00000248	0064A	U.950: .LONG	<<SMG\$MASK_ADR-U.950>-4>	:
		:TPASMASK		:
FFFF	0064E	U.951: .LONG	584	:
		:TPASTARGET		:
7185	00650	U.952: .WORD	-1	:
		:TPASTYPE		:
00000000*	00652	U.956: .WORD	29061	:
		:TPASADDR		:
0000022E	00656	U.957: .LONG	<<SMG\$MASK_ADR-U.957>-4>	:
		:TPASMASK		:
FFFF	0065A	U.958: .LONG	558	:
		:TPASTARGET		:
7186	0065C	U.959: .WORD	-1	:
		:TPASTYPE		:
00000000*	0065E	U.963: .WORD	29062	:
		:TPASADDR		:
00000252	00662	U.964: .LONG	<<SMG\$MASK_ADR-U.964>-4>	:
		:TPASMASK		:
FFFF	00666	U.965: .LONG	594	:
		:TPASTARGET		:
7187	00668	U.966: .WORD	-1	:
		:TPASTYPE		:
		U.970: .WORD	29063	:

00000000*	0066A	:TPASADDR					
		U.971:	LONG	<<SMG\$MASK_ADR-U.971>-4>		:	
0000022F	0066E	:TPASMASK				:	
		U.972:	LONG	559		:	
FFFF	00672	:TPASTARGET				:	
		U.973:	WORD	-1		:	
7188	00674	:TPASTYPE				:	
		U.977:	WORD	29064		:	
00000000*	00676	:TPASADDR				:	
		U.978:	LONG	<<SMG\$MASK_ADR-U.978>-4>		:	
00000230	0067A	:TPASMASK				:	
		U.979:	LONG	560		:	
FFFF	0067E	:TPASTARGET				:	
		U.980:	WORD	-1		:	
7189	00680	:TPASTYPE				:	
		U.984:	WORD	29065		:	
00000000*	00682	:TPASADDR				:	
		U.985:	LONG	<<SMG\$MASK_ADR-U.985>-4>		:	
00000231	00686	:TPASMASK				:	
		U.986:	LONG	561		:	
FFFF	0068A	:TPASTARGET				:	
		U.987:	WORD	-1		:	
718A	0068C	:TPASTYPE				:	
		U.991:	WORD	29066		:	
00000000*	0068E	:TPASADDR				:	
		U.992:	LONG	<<SMG\$MASK_ADR-U.992>-4>		:	
00000232	00692	:TPASMASK				:	
		U.993:	LONG	562		:	
FFFF	00696	:TPASTARGET				:	
		U.994:	WORD	-1		:	
718B	00698	:TPASTYPE				:	
		U.998:	WORD	29067		:	
00000000*	0069A	:TPASADDR				:	
		U.999:	LONG	<<SMG\$MASK_ADR-U.999>-4>		:	
00000233	0069E	:TPASMASK				:	
		U.1000:	LONG	563		:	
FFFF	006A2	:TPASTARGET				:	
		U.1001:	WORD	-1		:	
718C	006A4	:TPASTYPE				:	
		U.1005:	WORD	29068		:	
00000000*	006A6	:TPASADDR				:	
		U.1006:	LONG	<<SMG\$MASK_ADR-U.1006>-4>		:	
00000234	006AA	:TPASMASK				:	
		U.1007:	LONG	564		:	
FFFF	006AE	:TPASTARGET				:	
		U.1008:	WORD	-1		:	
718D	006B0	:TPASTYPE				:	
		U.1012:	WORD	29069		:	
00000000*	006B2	:TPASADDR				:	
		U.1013:	LONG	<<SMG\$MASK_ADR-U.1013>-4>		:	
00000235	006B6	:TPASMASK				:	
		U.1014:	LONG	565		:	
FFFF	006BA	:TPASTARGET				:	
		U.1015:	WORD	-1		:	
718E	006BC	:TPASTYPE				:	
		U.1019:	WORD	29070		:	
00000000*	006BE	:TPASADDR				:	

00000236	006C2	U.1020: .LONG	<<SMG\$MASK_ADR-U.1020>-4>	:
		:TP\$MASK		:
FFFF	006C6	U.1021: .LONG	566	:
		:TP\$TARGET		:
718F	006C8	U.1022: .WORD	-1	:
		:TP\$TYPE		:
00000000*	006CA	U.1026: .WORD	29071	:
		:TP\$ADDR		:
00000237	006CE	U.1027: .LONG	<<SMG\$MASK_ADR-U.1027>-4>	:
		:TP\$MASK		:
FFFF	006D2	U.1028: .LONG	567	:
		:TP\$TARGET		:
7190	006D4	U.1029: .WORD	-1	:
		:TP\$TYPE		:
00000000*	006D6	U.1033: .WORD	29072	:
		:TP\$ADDR		:
00000238	006DA	U.1034: .LONG	<<SMG\$MASK_ADR-U.1034>-4>	:
		:TP\$MASK		:
FFFF	006DE	U.1035: .LONG	568	:
		:TP\$TARGET		:
7191	006E0	U.1036: .WORD	-1	:
		:TP\$TYPE		:
00000000*	006E2	U.1040: .WORD	29073	:
		:TP\$ADDR		:
00000239	006E6	U.1041: .LONG	<<SMG\$MASK_ADR-U.1041>-4>	:
		:TP\$MASK		:
FFFF	006EA	U.1042: .LONG	569	:
		:TP\$TARGET		:
7192	006EC	U.1043: .WORD	-1	:
		:TP\$TYPE		:
00000000*	006EE	U.1047: .WORD	29074	:
		:TP\$ADDR		:
0000023A	006F2	U.1048: .LONG	<<SMG\$MASK_ADR-U.1048>-4>	:
		:TP\$MASK		:
FFFF	006F6	U.1049: .LONG	570	:
		:TP\$TARGET		:
7193	006F8	U.1050: .WORD	-1	:
		:TP\$TYPE		:
00000000*	006FA	U.1054: .WORD	29075	:
		:TP\$ADDR		:
0000023B	006FE	U.1055: .LONG	<<SMG\$MASK_ADR-U.1055>-4>	:
		:TP\$MASK		:
FFFF	00702	U.1056: .LONG	571	:
		:TP\$TARGET		:
7194	00704	U.1057: .WORD	-1	:
		:TP\$TYPE		:
00000000*	00706	U.1061: .WORD	29076	:
		:TP\$ADDR		:
0000023C	0070A	U.1062: .LONG	<<SMG\$MASK_ADR-U.1062>-4>	:
		:TP\$MASK		:
FFFF	0070E	U.1063: .LONG	572	:
		:TP\$TARGET		:
7195	00710	U.1064: .WORD	-1	:
		:TP\$TYPE		:
00000000*	00712	U.1068: .WORD	29077	:
		:TP\$ADDR		:
		U.1069: .LONG	<<SMG\$MASK_ADR-U.1069>-4>	:

0000023D	00716	:TPASMASK			
		U.1070: .LONG	573		:
FFFF	0071A	:TPASTARGET			:
		U.1071: .WORD	-1		:
7196	0071C	:TPASTYPE			:
		U.1075: .WORD	29078		:
00000000*	0071E	:TPASADDR			:
		U.1076: .LONG	<<SMG\$SMASK_ADR-U.1076>-4>		:
0000023E	00722	:TPASMASK			:
		U.1077: .LONG	574		:
FFFF	00726	:TPASTARGET			:
		U.1078: .WORD	-1		:
7197	00728	:TPASTYPE			:
		U.1082: .WORD	29079		:
00000000*	0072A	:TPASADDR			:
		U.1083: .LONG	<<SMG\$SMASK_ADR-U.1083>-4>		:
0000023F	0072E	:TPASMASK			:
		U.1084: .LONG	575		:
FFFF	00732	:TPASTARGET			:
		U.1085: .WORD	-1		:
7198	00734	:TPASTYPE			:
		U.1089: .WORD	29080		:
00000000*	00736	:TPASADDR			:
		U.1090: .LONG	<<SMG\$SMASK_ADR-U.1090>-4>		:
00000240	0073A	:TPASMASK			:
		U.1091: .LONG	576		:
FFFF	0073E	:TPASTARGET			:
		U.1092: .WORD	-1		:
7199	00740	:TPASTYPE			:
		U.1096: .WORD	29081		:
00000000*	00742	:TPASADDR			:
		U.1097: .LONG	<<SMG\$SMASK_ADR-U.1097>-4>		:
0000024E	00746	:TPASMASK			:
		U.1098: .LONG	590		:
FFFF	0074A	:TPASTARGET			:
		U.1099: .WORD	-1		:
719A	0074C	:TPASTYPE			:
		U.1103: .WORD	29082		:
00000000*	0074E	:TPASADDR			:
		U.1104: .LONG	<<SMG\$SMASK_ADR-U.1104>-4>		:
00000241	00752	:TPASMASK			:
		U.1105: .LONG	577		:
FFFF	00756	:TPASTARGET			:
		U.1106: .WORD	-1		:
719B	00758	:TPASTYPE			:
		U.1110: .WORD	29083		:
00000000*	0075A	:TPASADDR			:
		U.1111: .LONG	<<SMG\$SMASK_ADR-U.1111>-4>		:
00000242	0075E	:TPASMASK			:
		U.1112: .LONG	578		:
FFFF	00762	:TPASTARGET			:
		U.1113: .WORD	-1		:
719C	00764	:TPASTYPE			:
		U.1117: .WORD	29084		:
00000000*	00766	:TPASADDR			:
		U.1118: .LONG	<<SMG\$SMASK_ADR-U.1118>-4>		:
00000243	0076A	:TPASMASK			:

FFFF	0076E	U.1119: .LONG	579	:
		:TPASTARGET		:
719D	00770	U.1120: .WORD	-1	:
		:TPASTYPE		:
00000000*	00772	U.1124: .WORD	29085	:
		:TPASADDR		:
00000244	00776	U.1125: .LONG	<<SMG\$MASK_ADR-U.1125>-4>	:
		:TPASMASK		:
FFFF	0077A	U.1126: .LONG	580	:
		:TPASTARGET		:
719E	0077C	U.1127: .WORD	-1	:
		:TPASTYPE		:
00000000*	0077E	U.1131: .WORD	29086	:
		:TPASADDR		:
0000024D	00782	U.1132: .LONG	<<SMG\$MASK_ADR-U.1132>-4>	:
		:TPASMASK		:
FFFF	00786	U.1133: .LONG	589	:
		:TPASTARGET		:
719F	00788	U.1134: .WORD	-1	:
		:TPASTYPE		:
00000000*	0078A	U.1138: .WORD	29087	:
		:TPASADDR		:
00000245	0078E	U.1139: .LONG	<<SMG\$MASK_ADR-U.1139>-4>	:
		:TPASMASK		:
FFFF	00792	U.1140: .LONG	581	:
		:TPASTARGET		:
71A0	00794	U.1141: .WORD	-1	:
		:TPASTYPE		:
00000000*	00796	U.1145: .WORD	29088	:
		:TPASADDR		:
00000246	0079A	U.1146: .LONG	<<SMG\$MASK_ADR-U.1146>-4>	:
		:TPASMASK		:
FFFF	0079E	U.1147: .LONG	582	:
		:TPASTARGET		:
71A1	007A0	U.1148: .WORD	-1	:
		:TPASTYPE		:
00000000*	007A2	U.1152: .WORD	29089	:
		:TPASADDR		:
0000028B	007A6	U.1153: .LONG	<<SMG\$MASK_ADR-U.1153>-4>	:
		:TPASMASK		:
FFFF	007AA	U.1154: .LONG	651	:
		:TPASTARGET		:
71A2	007AC	U.1155: .WORD	-1	:
		:TPASTYPE		:
00000000*	007AE	U.1159: .WORD	29090	:
		:TPASADDR		:
0000028C	007B2	U.1160: .LONG	<<SMG\$MASK_ADR-U.1160>-4>	:
		:TPASMASK		:
FFFF	007B6	U.1161: .LONG	652	:
		:TPASTARGET		:
71A3	007B8	U.1162: .WORD	-1	:
		:TPASTYPE		:
00000000*	007BA	U.1166: .WORD	29091	:
		:TPASADDR		:
0000028D	007BE	U.1167: .LONG	<<SMG\$MASK_ADR-U.1167>-4>	:
		:TPASMASK		:
		U.1168: .LONG	653	:

FFFF	007C2	:TPASTARGET			
		U.1169: .WORD	-1		:
71A4	007C4	:TPASTYPE			:
		U.1173: .WORD	29092		:
00000000*	007C6	:TPASADDR			:
		U.1174: .LONG	<<SMG\$MASK_ADR-U.1174>-4>		:
0000028E	007CA	:TPASMASK			:
		U.1175: .LONG	654		:
FFFF	007CE	:TPASTARGET			:
		U.1176: .WORD	-1		:
71A5	007D0	:TPASTYPE			:
		U.1180: .WORD	29093		:
00000000*	007D2	:TPASADDR			:
		U.1181: .LONG	<<SMG\$MASK_ADR-U.1181>-4>		:
0000028F	007D6	:TPASMASK			:
		U.1182: .LONG	655		:
FFFF	007DA	:TPASTARGET			:
		U.1183: .WORD	-1		:
71A6	007DC	:TPASTYPE			:
		U.1187: .WORD	29094		:
00000000*	007DE	:TPASADDR			:
		U.1188: .LONG	<<SMG\$MASK_ADR-U.1188>-4>		:
00000290	007E2	:TPASMASK			:
		U.1189: .LONG	656		:
FFFF	007E6	:TPASTARGET			:
		U.1190: .WORD	-1		:
71A7	007E8	:TPASTYPE			:
		U.1194: .WORD	29095		:
00000000*	007EA	:TPASADDR			:
		U.1195: .LONG	<<SMG\$MASK_ADR-U.1195>-4>		:
00000291	007EE	:TPASMASK			:
		U.1196: .LONG	657		:
FFFF	007F2	:TPASTARGET			:
		U.1197: .WORD	-1		:
71A8	007F4	:TPASTYPE			:
		U.1201: .WORD	29096		:
00000000*	007F6	:TPASADDR			:
		U.1202: .LONG	<<SMG\$MASK_ADR-U.1202>-4>		:
00000292	007FA	:TPASMASK			:
		U.1203: .LONG	658		:
FFFF	007FE	:TPASTARGET			:
		U.1204: .WORD	-1		:
71A9	00800	:TPASTYPE			:
		U.1208: .WORD	29097		:
00000000*	00802	:TPASADDR			:
		U.1209: .LONG	<<SMG\$MASK_ADR-U.1209>-4>		:
00000293	00806	:TPASMASK			:
		U.1210: .LONG	659		:
FFFF	0080A	:TPASTARGET			:
		U.1211: .WORD	-1		:
71AA	0080C	:TPASTYPE			:
		U.1215: .WORD	29098		:
00000000*	0080E	:TPASADDR			:
		U.1216: .LONG	<<SMG\$MASK_ADR-U.1216>-4>		:
00000294	00812	:TPASMASK			:
		U.1217: .LONG	660		:
FFFF	00816	:TPASTARGET			:

15F6	00818	U.1218: .WORD	-1	:
		:TPASTYPE		:
FFFE	0081A	U.1219: .WORD	5622	:
		:TPASTARGET		:
	0081C	U.1220: .WORD	-2	:
		:EQUALS_STRING		:
99F8	0081C	U.27: .BLKB	0	:
		:TPASTYPE		:
0000*	0081E	U.1222: .WORD	-26120	:
		:TPASSUBEXP		:
00000000*	00820	U.1223: .WORD	<<U.3-U.1223>-2>	:
		:TPASACTION		:
0000*	00824	U.1224: .LONG	<<SMG\$\$NEXT_RECORD-U.1224>-4>	:
		:TPASTARGET		:
903D	00826	U.1225: .WORD	<<U.27-U.1225>-2>	:
		:TPASTYPE		:
00000000*	00828	U.1226: .WORD	-28611	:
		:TPASACTION		:
0000*	0082C	U.1227: .LONG	<<SMG\$\$STORE_CAP_MASK-U.1227>-4>	:
		:TPASTARGET		:
81F1	0082E	U.1229: .WORD	<<U.1228-U.1229>-2>	:
		:TPASTYPE		:
00000000*	00830	U.1230: .WORD	-32271	:
		:TPASACTION		:
85ED	00834	U.1231: .LONG	<<SMG\$\$SYNTAX_ERROR-U.1231>-4>	:
		:TPASTYPE		:
00000000*	00836	U.1232: .WORD	-31251	:
		:TPASACTION		:
	0083A	U.1233: .LONG	<<SMG\$\$SYNTAX_ERROR-U.1233>-4>	:
		:STRING_CAP_VALUE		:
99F8	0083A	U.1228: .BLKB	0	:
		:TPASTYPE		:
0000*	0083C	U.1234: .WORD	-26120	:
		:TPASSUBEXP		:
00000000*	0083E	U.1235: .WORD	<<U.3-U.1235>-2>	:
		:TPASACTION		:
0000*	00842	U.1236: .LONG	<<SMG\$\$NEXT_RECORD-U.1236>-4>	:
		:TPASTARGET		:
19F8	00844	U.1237: .WORD	<<U.1228-U.1237>-2>	:
		:TPASTYPE		:
0000*	00846	U.1238: .WORD	6648	:
		:TPASSUBEXP		:
0000*	00848	U.1240: .WORD	<<U.1239-U.1240>-2>	:
		:TPASTARGET		:
81F1	0084A	U.1241: .WORD	<<BEGIN_SCAN-U.1241>-2>	:
		:TPASTYPE		:
00000000*	0084C	U.1242: .WORD	-32271	:
		:TPASACTION		:
85ED	00850	U.1243: .LONG	<<SMG\$\$SYNTAX_ERROR-U.1243>-4>	:
		:TPASTYPE		:
00000000*	00852	U.1244: .WORD	-31251	:
		:TPASACTION		:
	00856	U.1245: .LONG	<<SMG\$\$SYNTAX_ERROR-U.1245>-4>	:
		:COMMA: .BLKB	0	:
902C	00856	:TPASTYPE		:
00000000*	00858	U.1246: .WORD	-28628	:
		:TPASACTION		:

FFFF	0085C	U.1247: .LONG	<<SMG\$BLANKS_OFF-U.1247>-4>	:
		:TPASTARGET		:
91F2	0085E	U.1248: .WORD	-1	:
		:TPASTYPE		:
00000000*	00860	U.1249: .WORD	-28174	:
		:TPASACTION		:
0000*	00864	U.1250: .LONG	<<SMG\$BLANKS_OFF-U.1250>-4>	:
		:TPASTARGET		:
99F8	00866	U.1251: .WORD	<<COMMA-U.1251>-2>	:
		:TPASTYPE		:
0000*	00868	U.1252: .WORD	-26120	:
		:TPASSUBEXP		:
00000000*	0086A	U.1253: .WORD	<<U.3-U.1253>-2>	:
		:TPASACTION		:
FFFF	0086E	U.1254: .LONG	<<SMG\$NEXT_RECORD-U.1254>-4>	:
		:TPASTARGET		:
85ED	00870	U.1255: .WORD	-1	:
		:TPASTYPE		:
00000000*	00872	U.1256: .WORD	-31251	:
		:TPASACTION		:
	00876	U.1257: .LONG	<<SMG\$SYNTAX_ERROR-U.1257>-4>	:
		:STRING_CAP		:
		U.1239: .BLKB	0	:
9027	00876	:TPASTYPE		:
		U.1258: .WORD	-28633	:
00000000*	00878	:TPASACTION		:
		U.1259: .LONG	<<SMG\$BLANKS_ON-U.1259>-4>	:
0000*	0087C	:TPASTARGET		:
		U.1261: .WORD	<<U.1260-U.1261>-2>	:
9022	0087E	:TPASTYPE		:
		U.1262: .WORD	-28638	:
00000000*	00880	:TPASACTION		:
		U.1263: .LONG	<<SMG\$BLANKS_ON-U.1263>-4>	:
0000*	00884	:TPASTARGET		:
		U.1265: .WORD	<<U.1264-U.1265>-2>	:
15ED	00886	:TPASTYPE		:
		U.1266: .WORD	5613	:
FFFE	00888	:TPASTARGET		:
		U.1267: .WORD	-2	:
	0088A	:SINGLE_QUOTE_STRING		:
		U.1260: .BLKB	0	:
19F8	0088A	:TPASTYPE		:
		U.1268: .WORD	6648	:
0000*	0088C	:TPASSUBEXP		:
		U.1270: .WORD	<<U.1269-U.1270>-2>	:
0000*	0088E	:TPASTARGET		:
		U.1271: .WORD	<<U.1260-U.1271>-2>	:
9024	00890	:TPASTYPE		:
		U.1272: .WORD	-28636	:
00000000V	00892	:TPASACTION		:
		U.1273: .LONG	<<CONVERT_ESCAPE-U.1273>-4>	:
0000*	00896	:TPASTARGET		:
		U.1274: .WORD	<<U.1260-U.1274>-2>	:
19F8	00898	:TPASTYPE		:
		U.1275: .WORD	6648	:
0000*	0089A	:TPASSUBEXP		:
		U.1277: .WORD	<<U.1276-U.1277>-2>	:

0000*	0089C	:TPASTARGET		
		U.1278: .WORD	<<U.1260-U.1278>-2>	:
19F8	0089E	:TPASTYPE		:
		U.1279: .WORD	6648	:
0000*	008A0	:TPASSUBEXP		:
		U.1281: .WORD	<<U.1280-U.1281>-2>	:
0000*	008A2	:TPASTARGET		:
		U.1282: .WORD	<<U.1260-U.1282>-2>	:
19F8	008A4	:TPASTYPE		:
		U.1283: .WORD	6648	:
0000*	008A6	:TPASSUBEXP		:
		U.1285: .WORD	<<U.1284-U.1285>-2>	:
0000*	008A8	:TPASTARGET		:
		U.1286: .WORD	<<U.1260-U.1286>-2>	:
9027	008AA	:TPASTYPE		:
		U.1287: .WORD	-28633	:
00000000*	008AC	:TPASACTION		:
		U.1288: .LONG	<<SMG\$SCOPY_CAP-U.1288>-4>	:
0000*	008B0	:TPASTARGET		:
		U.1289: .WORD	<<COMMA-U.1289>-2>	:
91F2	008B2	:TPASTYPE		:
		U.1290: .WORD	-28174	:
00000000*	008B4	:TPASACTION		:
		U.1291: .LONG	<<SMG\$SSAVE_TOKEN_STRING-U.1291>-4>	:
0000*	008B8	:TPASTARGET		:
		U.1292: .WORD	<<U.1260-U.1292>-2>	:
8022	008BA	:TPASTYPE		:
		U.1293: .WORD	-32734	:
00000000*	008BC	:TPASACTION		:
		U.1294: .LONG	<<SMG\$SSYNTAX_ERROR-U.1294>-4>	:
91ED	008C0	:TPASTYPE		:
		U.1295: .WORD	-28179	:
00000000*	008C2	:TPASACTION		:
		U.1296: .LONG	<<SMG\$SSAVE_TOKEN_STRING-U.1296>-4>	:
0000*	008C6	:TPASTARGET		:
		U.1297: .WORD	<<U.1260-U.1297>-2>	:
15F6	008C8	:TPASTYPE		:
		U.1298: .WORD	5622	:
FFFF	008CA	:TPASTARGET		:
		U.1299: .WORD	-1	:
	008CC	:DOUBLE_QUOTE_STRING		:
		U.1264: .BLKB	0	:
19F8	008CC	:TPASTYPE		:
		U.1300: .WORD	6648	:
0000*	008CE	:TPASSUBEXP		:
		U.1301: .WORD	<<U.1269-U.1301>-2>	:
0000*	008D0	:TPASTARGET		:
		U.1302: .WORD	<<U.1264-U.1302>-2>	:
9024	008D2	:TPASTYPE		:
		U.1303: .WORD	-28636	:
00000000V	008D4	:TPASACTION		:
		U.1304: .LONG	<<CONVERT_ESCAPE-U.1304>-4>	:
0000*	008D8	:TPASTARGET		:
		U.1305: .WORD	<<U.1264-U.1305>-2>	:
19F8	008DA	:TPASTYPE		:
		U.1306: .WORD	6648	:
0000*	008DC	:TPASSUBEXP		:

0000*	008DE	U.1307: .WORD	<<U.1276-U.1307>-2>	:
		:TPASTARGET		:
19F8	008E0	U.1308: .WORD	<<U.1264-U.1308>-2>	:
		:TPASTYPE		:
0000*	008E2	U.1309: .WORD	6648	:
		:TPASSUBEXP		:
0000*	008E4	U.1310: .WORD	<<U.1280-U.1310>-2>	:
		:TPASTARGET		:
19F8	008E6	U.1311: .WORD	<<U.1264-U.1311>-2>	:
		:TPASTYPE		:
0000*	008E8	U.1312: .WORD	6648	:
		:TPASSUBEXP		:
0000*	008EA	U.1313: .WORD	<<U.1284-U.1313>-2>	:
		:TPASTARGET		:
9022	008EC	U.1314: .WORD	<<U.1264-U.1314>-2>	:
		:TPASTYPE		:
00000000*	008EE	U.1315: .WORD	-28638	:
		:TPASACTION		:
0000*	008F2	U.1316: .LONG	<<SMG\$\$\$COPY_CAP-U.1316>-4>	:
		:TPASTARGET		:
91F2	008F4	U.1317: .WORD	<<COMMA-U.1317>-2>	:
		:TPASTYPE		:
00000000*	008F6	U.1318: .WORD	-28174	:
		:TPASACTION		:
0000*	008FA	U.1319: .LONG	<<SMG\$\$\$SAVE_TOKEN_STRING-U.1319>-4>	:
		:TPASTARGET		:
8027	008FC	U.1320: .WORD	<<U.1264-U.1320>-2>	:
		:TPASTYPE		:
00000000*	008FE	U.1321: .WORD	-32729	:
		:TPASACTION		:
91ED	00902	U.1322: .LONG	<<SMG\$\$\$SYNTAX_ERROR-U.1322>-4>	:
		:TPASTYPE		:
00000000*	00904	U.1323: .WORD	-28179	:
		:TPASACTION		:
0000*	00908	U.1324: .LONG	<<SMG\$\$\$SAVE_TOKEN_STRING-U.1324>-4>	:
		:TPASTARGET		:
15F6	0090A	U.1325: .WORD	<<U.1264-U.1325>-2>	:
		:TPASTYPE		:
FFFF	0090C	U.1326: .WORD	5622	:
		:TPASTARGET		:
		U.1327: .WORD	-1	:
	0090E	:UNDERSCORE		:
		U.1269: .BLKB	0	:
105F	0090E	:TPASTYPE		:
0000*	00910	U.1328: .WORD	4191	:
		:TPASTARGET		:
15F6	00912	U.1330: .WORD	<<U.1329-U.1330>-2>	:
		:TPASTYPE		:
FFFE	00914	U.1331: .WORD	5622	:
		:TPASTARGET		:
		U.1332: .WORD	-2	:
	00916	:UNDERSCORE2		:
		U.1329: .BLKB	0	:
9024	00916	:TPASTYPE		:
00000000V	00918	U.1333: .WORD	-28636	:
		:TPASACTION		:
		U.1334: .LONG	<<INSERT_DOLLAR-U.1334>-4>	:

FFFF	0091C	:TPASTARGET		
		U.1335: .WORD	-1	:
905E	0091E	:TPASTYPE		:
		U.1336: .WORD	-28578	:
00000000V	00920	:TPASACTION		:
		U.1337: .LONG	<<INSERT_CARROT-U.1337>-4>	:
FFFF	00924	:TPASTARGET		:
		U.1338: .WORD	-1	:
9021	00926	:TPASTYPE		:
		U.1339: .WORD	-28639	:
00000000V	00928	:TPASACTION		:
		U.1340: .LONG	<<INSERT_EXCLAMATION-U.1340>-4>	:
FFFF	0092C	:TPASTARGET		:
		U.1341: .WORD	-1	:
9028	0092E	:TPASTYPE		:
		U.1342: .WORD	-28632	:
00000000V	00930	:TPASACTION		:
		U.1343: .LONG	<<INSERT_PARENTHESIS-U.1343>-4>	:
FFFF	00934	:TPASTARGET		:
		U.1344: .WORD	-1	:
15F6	00936	:TPASTYPE		:
		U.1345: .WORD	5622	:
FFFE	00938	:TPASTARGET		:
		U.1346: .WORD	-2	:
	0093A	:CARROT		:
		U.1276: .BLKB	0	:
105E	0093A	:TPASTYPE		:
		U.1347: .WORD	4190	:
0000*	0093C	:TPASTARGET		:
		U.1349: .WORD	<<U.1348-U.1349>-2>	:
15F6	0093E	:TPASTYPE		:
		U.1350: .WORD	5622	:
FFFE	00940	:TPASTARGET		:
		U.1351: .WORD	-2	:
	00942	:CONTROL_CHAR		:
		U.1348: .BLKB	0	:
95ED	00942	:TPASTYPE		:
		U.1352: .WORD	-27155	:
00000000*	00944	:TPASACTION		:
		U.1353: .LONG	<<CONVERT_CONTROL-U.1353>-4>	:
FFFF	00948	:TPASTARGET		:
		U.1354: .WORD	-1	:
	0094A	:FAO_STRING		:
		U.1280: .BLKB	0	:
8021	0094A	:TPASTYPE		:
		U.1355: .WORD	-32735	:
00000000*	0094C	:TPASACTION		:
		U.1356: .LONG	<<SMG\$\$\$SAVE_TOKEN_STRING-U.1356>-4>	:
15F6	00950	:TPASTYPE		:
		U.1357: .WORD	5622	:
FFFE	00952	:TPASTARGET		:
		U.1358: .WORD	-2	:
8055	00954	:TPASTYPE		:
		U.1359: .WORD	-32683	:
00000000*	00956	:TPASACTION		:
		U.1360: .LONG	<<SMG\$\$\$SAVE_TOKEN_STRING-U.1360>-4>	:
91F1	0095A	:TPASTYPE		:

00000000V	0095C	U.1361: .WORD	-28175	:
		:TPASACTION		:
FFFE	00960	U.1362: .LONG	<<INVALID_DIRECTIVE-U.1362>-4>	:
		:TPASTARGET		:
95ED	00962	U.1363: .WORD	-2	:
		:TPASTYPE		:
00000000V	00964	U.1364: .WORD	-27155	:
		:TPASACTION		:
FFFE	00968	U.1365: .LONG	<<INVALID_DIRECTIVE-U.1365>-4>	:
		:TPASTARGET		:
F04C	0096A	U.1366: .WORD	-2	:
		:TPASTYPE		:
00000000*	0096C	U.1367: .WORD	-4020	:
		:TPASACTION		:
00000000*	00970	U.1368: .LONG	<<SMG\$\$\$SAVE_TOKEN_STRING-U.1368>-4>	:
		:TPASADDR		:
FFFFFFFF	00974	U.1369: .LONG	<<SMG\$\$\$MASK_ADR-U.1369>-4>	:
		:TPASMASK		:
FFFF	00978	U.1370: .LONG	-1	:
		:TPASTARGET		:
91F1	0097A	U.1371: .WORD	-1	:
		:TPASTYPE		:
00000000V	0097C	U.1372: .WORD	-28175	:
		:TPASACTION		:
FFFE	00980	U.1373: .LONG	<<INVALID_DIRECTIVE-U.1373>-4>	:
		:TPASTARGET		:
95ED	00982	U.1374: .WORD	-2	:
		:TPASTYPE		:
00000000V	00984	U.1375: .WORD	-27155	:
		:TPASACTION		:
FFFE	00988	U.1376: .LONG	<<INVALID_DIRECTIVE-U.1376>-4>	:
		:TPASTARGET		:
	0098A	U.1377: .WORD	-2	:
		:ARITH_CAP		:
9028	0098A	U.1284: .BLKB	0	:
		:TPASTYPE		:
00000000V	0098C	U.1378: .WORD	-28632	:
		:TPASACTION		:
0000*	00990	U.1379: .LONG	<<INIT_ARITH_CAP-U.1379>-4>	:
		:TPASTARGET		:
11F2	00992	U.1381: .WORD	<<U.1380-U.1381>-2>	:
		:TPASTYPE		:
0000*	00994	U.1382: .WORD	4594	:
		:TPASTARGET		:
15F6	00996	U.1383: .WORD	<<U.1284-U.1383>-2>	:
		:TPASTYPE		:
FFFE	00998	U.1384: .WORD	5622	:
		:TPASTARGET		:
	0099A	U.1385: .WORD	-2	:
		:ARITH_CAP2		:
19F8	0099A	U.1380: .BLKB	0	:
		:TPASTYPE		:
0000*	0099C	U.1386: .WORD	6648	:
		:TPASSUBEXP		:
FFFF	0099E	U.1388: .WORD	<<U.1387-U.1388>-2>	:
		:TPASTARGET		:
		U.1389: .WORD	-1	:

15F6	009A0	:TPASTYPE	
		U.1390: .WORD	5622
FFFE	009A2	:TPASTARGET	
		U.1391: .WORD	-2
	009A4	:OPERATOR:	
		.BLKB	0
702B	009A4	:TPASTYPE	
		U.1392: .WORD	28715
00000000*	009A6	:TPASADDR	
		U.1393: .LONG	<<SMG\$MASK_ADR-U.1393>-4>
FFFFFFFFB	009AA	:TPASMASK	
		U.1394: .LONG	-5
0000*	009AE	:TPASTARGET	
		U.1395: .WORD	<<U.1387-U.1395>-2>
702D	009B0	:TPASTYPE	
		U.1396: .WORD	28717
00000000*	009B2	:TPASADDR	
		U.1397: .LONG	<<SMG\$MASK_ADR-U.1397>-4>
FFFFFFFFA	009B6	:TPASMASK	
		U.1398: .LONG	-6
0000*	009BA	:TPASTARGET	
		U.1399: .WORD	<<U.1387-U.1399>-2>
702A	009BC	:TPASTYPE	
		U.1400: .WORD	28714
00000000*	009BE	:TPASADDR	
		U.1401: .LONG	<<SMG\$MASK_ADR-U.1401>-4>
FFFFFFFF9	009C2	:TPASMASK	
		U.1402: .LONG	-7
0000*	009C6	:TPASTARGET	
		U.1403: .WORD	<<U.1387-U.1403>-2>
702F	009C8	:TPASTYPE	
		U.1404: .WORD	28719
00000000*	009CA	:TPASADDR	
		U.1405: .LONG	<<SMG\$MASK_ADR-U.1405>-4>
FFFFFFFF8	009CE	:TPASMASK	
		U.1406: .LONG	-8
0000*	009D2	:TPASTARGET	
		U.1407: .WORD	<<U.1387-U.1407>-2>
9029	009D4	:TPASTYPE	
		U.1408: .WORD	-28631
00000000V	009D6	:TPASACTION	
		U.1409: .LONG	<<END_ARITH_CAP-U.1409>-4>
FFFF	009DA	:TPASTARGET	
		U.1410: .WORD	-1
85ED	009DC	:TPASTYPE	
		U.1411: .WORD	-31251
00000000V	009DE	:TPASACTION	
		U.1412: .LONG	<<EXPRESSION_ERROR-U.1412>-4>
	009E2	:OPERAND	
		U.1387: .BLKB	0
91F3	009E2	:TPASTYPE	
		U.1413: .WORD	-28173
00000000V	009E4	:TPASACTION	
		U.1414: .LONG	<<STORE_OPERAND-U.1414>-4>
0000*	009E8	:TPASTARGET	
		U.1415: .WORD	<<OPERATOR-U.1415>-2>
19F8	009EA	:TPASTYPE	

```

0000* 009EC :TPASSUBEXP      6648
0000* 009EE :TPASTARGET      <<U.1417-U.1418>-2>
81F1 009F0 :TPASTYPE         <<OPERATOR-U.1419>-2>
00000000V 009F2 :TPASACTION   -32271
85ED 009F6 :TPASTYPE         <<EXPRESSION_ERROR-U.1421>-4>
00000000V 009F8 :TPASACTION   -31251
009FC :SUBSTITUTION         <<EXPRESSION_ERROR-U.1423>-4>
0025 009FC :TPASTYPE         0
15F6 009FE :TPASTYPE         37
FFFE 00A00 :TPASTARGET      5622
91F3 00A02 :TPASTYPE         -2
00000000V 00A04 :TPASACTION   -28173
FFFF 00A08 :TPASTARGET      <<STORE_SUBSTITUTION-U.1428>-4>
85ED 00A0A :TPASTYPE         -1
00000000V 00A0C :TPASACTION   -31251
U.1431: .LONG <<EXPRESSION_ERROR-U.1431>-4>

```

.PSECT _LIB\$KEY0\$,NOWRT, SHR, PIC,1

```

00000 SMG$SA_STRING_KEYWDS::
00000 :TPASKEY0 BLKB 0
0000* 00000 :TPASKEY U.1: .BLKB 0
0000* 00002 :TPASKEY U.30: .WORD <U.29-U.1>
0000* 00004 :TPASKEY U.35: .WORD <U.34-U.1>
0000* 00006 :TPASKEY U.40: .WORD <U.39-U.1>
0000* 00008 :TPASKEY U.45: .WORD <U.44-U.1>
0000* 0000A :TPASKEY U.50: .WORD <U.49-U.1>
0000* 0000C :TPASKEY U.56: .WORD <U.55-U.1>
0000* 0000E :TPASKEY U.65: .WORD <U.64-U.1>
0000* 00010 :TPASKEY U.72: .WORD <U.71-U.1>
0000* 00012 :TPASKEY U.79: .WORD <U.78-U.1>

```

```
TPARSE tables for string capabilities
CONVERT_CONTROL - Convert ^ to control character
```

14-Sep-1984 13:10:04

VAX-11 B11ss-32 V4.0-742
[SMGRTL.SRC]SMGSTRTAB.B32:1

Page 61
(4)

Address	Content	Label
0000* 00014	U.86: .WORD	<U.85-U.1>
0000* 00016	:TPASKEY	
0000* 00016	U.93: .WORD	<U.92-U.1>
0000* 00018	:TPASKEY	
0000* 00018	U.100: .WORD	<U.99-U.1>
0000* 0001A	:TPASKEY	
0000* 0001A	U.107: .WORD	<U.106-U.1>
0000* 0001C	:TPASKEY	
0000* 0001C	U.114: .WORD	<U.113-U.1>
0000* 0001E	:TPASKEY	
0000* 0001E	U.121: .WORD	<U.120-U.1>
0000* 00020	:TPASKEY	
0000* 00020	U.128: .WORD	<U.127-U.1>
0000* 00022	:TPASKEY	
0000* 00022	U.135: .WORD	<U.134-U.1>
0000* 00024	:TPASKEY	
0000* 00024	U.142: .WORD	<U.141-U.1>
0000* 00026	:TPASKEY	
0000* 00026	U.149: .WORD	<U.148-U.1>
0000* 00028	:TPASKEY	
0000* 00028	U.156: .WORD	<U.155-U.1>
0000* 0002A	:TPASKEY	
0000* 0002A	U.163: .WORD	<U.162-U.1>
0000* 0002C	:TPASKEY	
0000* 0002C	U.170: .WORD	<U.169-U.1>
0000* 0002E	:TPASKEY	
0000* 0002E	U.177: .WORD	<U.176-U.1>
0000* 00030	:TPASKEY	
0000* 00030	U.184: .WORD	<U.183-U.1>
0000* 00032	:TPASKEY	
0000* 00032	U.191: .WORD	<U.190-U.1>
0000* 00034	:TPASKEY	
0000* 00034	U.198: .WORD	<U.197-U.1>
0000* 00036	:TPASKEY	
0000* 00036	U.205: .WORD	<U.204-U.1>
0000* 00038	:TPASKEY	
0000* 00038	U.212: .WORD	<U.211-U.1>
0000* 0003A	:TPASKEY	
0000* 0003A	U.219: .WORD	<U.218-U.1>
0000* 0003C	:TPASKEY	
0000* 0003C	U.226: .WORD	<U.225-U.1>
0000* 0003E	:TPASKEY	
0000* 0003E	U.233: .WORD	<U.232-U.1>
0000* 00040	:TPASKEY	
0000* 00040	U.240: .WORD	<U.239-U.1>
0000* 00042	:TPASKEY	
0000* 00042	U.247: .WORD	<U.246-U.1>
0000* 00044	:TPASKEY	
0000* 00044	U.254: .WORD	<U.253-U.1>
0000* 00046	:TPASKEY	
0000* 00046	U.261: .WORD	<U.260-U.1>
0000* 00048	:TPASKEY	
0000* 00048	U.268: .WORD	<U.267-U.1>
0000* 0004A	:TPASKEY	
0000* 0004A	U.275: .WORD	<U.274-U.1>
0000* 0004C	:TPASKEY	
0000* 0004C	U.282: .WORD	<U.281-U.1>

0000*	0004C	:TPASKEY	
		U.289:	.WORD <U.288-U.1>
0000*	0004E	:TPASKEY	
		U.296:	.WORD <U.295-U.1>
0000*	00050	:TPASKEY	
		U.303:	.WORD <U.302-U.1>
0000*	00052	:TPASKEY	
		U.310:	.WORD <U.309-U.1>
0000*	00054	:TPASKEY	
		U.317:	.WORD <U.316-U.1>
0000*	00056	:TPASKEY	
		U.324:	.WORD <U.323-U.1>
0000*	00058	:TPASKEY	
		U.331:	.WORD <U.330-U.1>
0000*	0005A	:TPASKEY	
		U.338:	.WORD <U.337-U.1>
0000*	0005C	:TPASKEY	
		U.345:	.WORD <U.344-U.1>
0000*	0005E	:TPASKEY	
		U.352:	.WORD <U.351-U.1>
0000*	00060	:TPASKEY	
		U.359:	.WORD <U.358-U.1>
0000*	00062	:TPASKEY	
		U.366:	.WORD <U.365-U.1>
0000*	00064	:TPASKEY	
		U.373:	.WORD <U.372-U.1>
0000*	00066	:TPASKEY	
		U.380:	.WORD <U.379-U.1>
0000*	00068	:TPASKEY	
		U.387:	.WORD <U.386-U.1>
0000*	0006A	:TPASKEY	
		U.394:	.WORD <U.393-U.1>
0000*	0006C	:TPASKEY	
		U.401:	.WORD <U.400-U.1>
0000*	0006E	:TPASKEY	
		U.408:	.WORD <U.407-U.1>
0000*	00070	:TPASKEY	
		U.415:	.WORD <U.414-U.1>
0000*	00072	:TPASKEY	
		U.422:	.WORD <U.421-U.1>
0000*	00074	:TPASKEY	
		U.429:	.WORD <U.428-U.1>
0000*	00076	:TPASKEY	
		U.436:	.WORD <U.435-U.1>
0000*	00078	:TPASKEY	
		U.443:	.WORD <U.442-U.1>
0000*	0007A	:TPASKEY	
		U.450:	.WORD <U.449-U.1>
0000*	0007C	:TPASKEY	
		U.457:	.WORD <U.456-U.1>
0000*	0007E	:TPASKEY	
		U.464:	.WORD <U.463-U.1>
0000*	00080	:TPASKEY	
		U.471:	.WORD <U.470-U.1>
0000*	00082	:TPASKEY	
		U.478:	.WORD <U.477-U.1>
0000*	00084	:TPASKEY	

100

0000*	000BE	:TPASKEY	
		U.688:	.WORD <U.687-U.1>
0000*	000C0	:TPASKEY	
		U.695:	.WORD <U.694-U.1>
0000*	000C2	:TPASKEY	
		U.702:	.WORD <U.701-U.1>
0000*	000C4	:TPASKEY	
		U.709:	.WORD <U.708-U.1>
0000*	000C6	:TPASKEY	
		U.716:	.WORD <U.715-U.1>
0000*	000C8	:TPASKEY	
		U.723:	.WORD <U.722-U.1>
0000*	000CA	:TPASKEY	
		U.730:	.WORD <U.729-U.1>
0000*	000CC	:TPASKEY	
		U.737:	.WORD <U.736-U.1>
0000*	000CE	:TPASKEY	
		U.744:	.WORD <U.743-U.1>
0000*	000D0	:TPASKEY	
		U.751:	.WORD <U.750-U.1>
0000*	000D2	:TPASKEY	
		U.758:	.WORD <U.757-U.1>
0000*	000D4	:TPASKEY	
		U.765:	.WORD <U.764-U.1>
0000*	000D6	:TPASKEY	
		U.772:	.WORD <U.771-U.1>
0000*	000D8	:TPASKEY	
		U.779:	.WORD <U.778-U.1>
0000*	000DA	:TPASKEY	
		U.786:	.WORD <U.785-U.1>
0000*	000DC	:TPASKEY	
		U.793:	.WORD <U.792-U.1>
0000*	000DE	:TPASKEY	
		U.800:	.WORD <U.799-U.1>
0000*	000E0	:TPASKEY	
		U.807:	.WORD <U.806-U.1>
0000*	000E2	:TPASKEY	
		U.814:	.WORD <U.813-U.1>
0000*	000E4	:TPASKEY	
		U.821:	.WORD <U.820-U.1>
0000*	000E6	:TPASKEY	
		U.828:	.WORD <U.827-U.1>
0000*	000E8	:TPASKEY	
		U.835:	.WORD <U.834-U.1>
0000*	000EA	:TPASKEY	
		U.842:	.WORD <U.841-U.1>
0000*	000EC	:TPASKEY	
		U.849:	.WORD <U.848-U.1>
0000*	000EE	:TPASKEY	
		U.856:	.WORD <U.855-U.1>
0000*	000F0	:TPASKEY	
		U.863:	.WORD <U.862-U.1>
0000*	000F2	:TPASKEY	
		U.870:	.WORD <U.869-U.1>
0000*	000F4	:TPASKEY	
		U.877:	.WORD <U.876-U.1>
0000*	000F6	:TPASKEY	

Address	Content	Comment
0000* 000F8	U.884: .WORD ;TPASKEY	<U.883-U.1>
0000* 000FA	U.891: .WORD ;TPASKEY	<U.890-U.1>
0000* 000FC	U.898: .WORD ;TPASKEY	<U.897-U.1>
0000* 000FE	U.905: .WORD ;TPASKEY	<U.904-U.1>
0000* 00100	U.912: .WORD ;TPASKEY	<U.911-U.1>
0000* 00102	U.919: .WORD ;TPASKEY	<U.918-U.1>
0000* 00104	U.926: .WORD ;TPASKEY	<U.925-U.1>
0000* 00106	U.933: .WORD ;TPASKEY	<U.932-U.1>
0000* 00108	U.940: .WORD ;TPASKEY	<U.939-U.1>
0000* 0010A	U.947: .WORD ;TPASKEY	<U.946-U.1>
0000* 0010C	U.954: .WORD ;TPASKEY	<U.953-U.1>
0000* 0010E	U.961: .WORD ;TPASKEY	<U.960-U.1>
0000* 00110	U.968: .WORD ;TPASKEY	<U.967-U.1>
0000* 00112	U.975: .WORD ;TPASKEY	<U.974-U.1>
0000* 00114	U.982: .WORD ;TPASKEY	<U.981-U.1>
0000* 00116	U.989: .WORD ;TPASKEY	<U.988-U.1>
0000* 00118	U.996: .WORD ;TPASKEY	<U.995-U.1>
0000* 0011A	U.1003: .WORD ;TPASKEY	<U.1002-U.1>
0000* 0011C	U.1010: .WORD ;TPASKEY	<U.1009-U.1>
0000* 0011E	U.1017: .WORD ;TPASKEY	<U.1016-U.1>
0000* 00120	U.1024: .WORD ;TPASKEY	<U.1023-U.1>
0000* 00122	U.1031: .WORD ;TPASKEY	<U.1030-U.1>
0000* 00124	U.1038: .WORD ;TPASKEY	<U.1037-U.1>
0000* 00126	U.1045: .WORD ;TPASKEY	<U.1044-U.1>
0000* 00128	U.1052: .WORD ;TPASKEY	<U.1051-U.1>
0000* 0012A	U.1059: .WORD ;TPASKEY	<U.1058-U.1>
0000* 0012C	U.1066: .WORD ;TPASKEY	<U.1065-U.1>
0000* 0012E	U.1073: .WORD ;TPASKEY	<U.1072-U.1>
	U.1080: .WORD	<U.1079-U.1>

```

0000* 00130 ;TPASKEY
           U.1087: .WORD <U.1086-U.1>
0000* 00132 ;TPASKEY
           U.1094: .WORD <U.1093-U.1>
0000* 00134 ;TPASKEY
           U.1101: .WORD <U.1100-U.1>
0000* 00136 ;TPASKEY
           U.1108: .WORD <U.1107-U.1>
0000* 00138 ;TPASKEY
           U.1115: .WORD <U.1114-U.1>
0000* 0013A ;TPASKEY
           U.1122: .WORD <U.1121-U.1>
0000* 0013C ;TPASKEY
           U.1129: .WORD <U.1128-U.1>
0000* 0013E ;TPASKEY
           U.1136: .WORD <U.1135-U.1>
0000* 00140 ;TPASKEY
           U.1143: .WORD <U.1142-U.1>
0000* 00142 ;TPASKEY
           U.1150: .WORD <U.1149-U.1>
0000* 00144 ;TPASKEY
           U.1157: .WORD <U.1156-U.1>
0000* 00146 ;TPASKEY
           U.1164: .WORD <U.1163-U.1>
0000* 00148 ;TPASKEY
           U.1171: .WORD <U.1170-U.1>
0000* 0014A ;TPASKEY
           U.1178: .WORD <U.1177-U.1>
0000* 0014C ;TPASKEY
           U.1185: .WORD <U.1184-U.1>
0000* 0014E ;TPASKEY
           U.1192: .WORD <U.1191-U.1>
0000* 00150 ;TPASKEY
           U.1199: .WORD <U.1198-U.1>
0000* 00152 ;TPASKEY
           U.1206: .WORD <U.1205-U.1>
0000* 00154 ;TPASKEY
           U.1213: .WORD <U.1212-U.1>

```

```

.EXTRN OTSSCVT TI L, SMG$$BLANKS_OFF
.EXTRN SMG$$BLANKS_ON, SMG$$COPY_CAP
.EXTRN SMG$$FLUSH_ARITHMETIC
.EXTRN SMG$$FLUSH_SAVED_BUFFER
.EXTRN SMG$$MISSING_END
.EXTRN SMG$$NEXT_RECORD
.EXTRN SMG$$SAVE_TOKEN_STRING
.EXTRN SMG$$STORE_CAP_MASK
.EXTRN SMG$$SYNTAX_ERROR
.EXTRN SMG$_ERRAT_CIN, SMG$_INVDIR
.EXTRN SMG$_INVEXP, SMG$_MISTERNAM
.EXTRN SMG$_NOTSTRCAP, SMG$_SYNERR
.EXTRN SMG$$MASK_ADR, SMG$$NEXT_NEGATIVE_NUMBER
.EXTRN SMG$$CURRENT_LINE
.EXTRN SMG$$DATA_OFFSET

.PSECT _SMG$CODE, NOWRT, SHR, PIC, 2

```


SMG\$STRING_TABL TPARSE tables for string capabilities
1-003 CONVERT_CONTROL - Convert ^ to control

M 13
16-Sep-1984 01:22:35
14-Sep-1984 13:10:04

VAX-11 Bliss-32 V4.0-742
[SMGRTL.SRC]SMGSTRTAB.B32;1

Page 67
(4)

0004 00000 CONVERT_CONTROL:

	52	00000000G	00	9E	00002	.WORD	Save R2	0613
	5E		04	C2	00009	MOVAB	SMG\$\$FLUSH_SAVED_BUFFER, R2	
		48	AC	D5	0000C	SUBL2	#4, SP	0662
			0D	12	0000F	TSTL	72(AP)	
		00000000G	00	9F	00011	BNEQ	1\$	0664
	00000000G	00	01	FB	00017	PUSHAB	SMG\$ MISTERNAM	
						CALLS	#1, [IB\$STOP	
		14	BC	90	0001E	1\$: MOVB	220(AP), CONTROL_CHAR	0674
6E		40	8F	8B	00022	BICB3	#64, CONTROL_CHAR, MOVE_CHAR	0680
			6C	FA	00027	CALLG	(AP), SMG\$\$FLUSH_SAVED_BUFFER	0688
	54	AC	01	D0	0002A	MOVL	#1, 84(AP)	0690
	58	AC	6E	9E	0002E	MOVAB	MOVE_CHAR, 88(AP)	0691
		62	6C	FA	00032	CALLG	(AP), SMG\$\$FLUSH_SAVED_BUFFER	0693
		50	01	D0	00035	MOVL	#1, R0	0695
				04	00038	RET		0697

: Routine Size: 57 bytes, Routine Base: _SMG\$CODE + 0000

```

625 0698 1 XSBTTL 'CONVERT_ESCAPE - Convert $ to escape character'
626 0699 1 ROUTINE CONVERT_ESCAPE =
627 0700 1
628 0701 1
629 0702 1
630 0703 1
631 0704 1
632 0705 1
633 0706 1
634 0707 1
635 0708 1
636 0709 1
637 0710 1
638 0711 1
639 0712 1
640 0713 1
641 0714 1
642 0715 1
643 0716 1
644 0717 1
645 0718 1
646 0719 1
647 0720 1
648 0721 1
649 0722 1
650 0723 1
651 0724 1
652 0725 1
653 0726 1
654 0727 1
655 0728 1
656 0729 1
657 0730 1
658 0731 2
659 0732 2
660 0733 2
661 0734 2
662 0735 2
663 0736 2
664 0737 2
665 0738 2
666 0739 2
667 0740 2
668 0741 2
669 0742 2
670 0743 2
671 0744 2
672 0745 2
673 0746 2
674 0747 2
675 0748 2
676 0749 2
677 0750 2
678 0751 2
679 0752 2
680 0753 2
681 0754 2

  **
  FUNCTIONAL DESCRIPTION:
      Stores an escape character in the current capability string in
      TERMTABLE.EXE.

  CALLING SEQUENCE:
      status = CONVERT_ESCAPE ()

  FORMAL PARAMETERS:
      NONE

  IMPLICIT INPUTS:
      AP      Points to TPARSE parameter block

  IMPLICIT OUTPUTS:
      NONE

  COMPLETION STATUS:
      SSS_NORMAL

  SIDE EFFECTS:
      --

  BEGIN
  BUILTIN
  CALLG,
  AP;
  MAP
  AP : REF BLOCK [,BYTE];
  LITERAL
  K_ESCAPE = %X'1B';
  BIND
  CAP_PTRS = .AP [PARAM_L_CUR_TERM_DEF] : VECTOR [,WORD];

  + The NAME capability should have preceeded this.  Complain if it didn't.
  -
      IF CAP_PTRS EQL 0
      THEN
        SIGNAL_STOP (SMG$_MISTERNAM);

  + Move an escape character into the capability string instead of
  the $.
  -
  Part of the string may already be copied - append in this part so

```

```

: 682      0755 2 | as not to overwrite it.
: 683      0756 2 |
: 684      0757 2 |
: 685      0758 2 | CALLG (.AP, SMG$$FLUSH_SAVED_BUFFER);
: 686      0759 2 |
: 687      0760 2 |
: 688      0761 2 | +
: 689      0762 2 | Insert the escape character.
: 690      0763 2 |
: 691      0764 2 | AP [PARAM_L_SAVED_TOKENCNT] = 1;
: 692      0765 2 | AP [PARAM_L_SAVED_TOKENSTR] = UPLIT (BYTE (K_ESCAPE));
: 693      0766 2 | | escape now 'saved'
: 694      0767 2 | CALLG (.AP, SMG$$FLUSH_SAVED_BUFFER);
: 695      0768 2 |
: 696      0769 2 | RETURN (SS$_NORMAL);
: 697      0770 2 |
: 698      0771 1 | END;

```

! end of routine CONVERT_ESCAPE

00039 .BLKB 3
1B 0003C P.AAA: .BYTE 27

				0004 00000 CONVERT_ESCAPE:			
	52	00000000G	00	9E 00002	.WORD	Save R2	0699
		48	AC	D5 00009	MOVAB	SMG\$\$FLUSH_SAVED_BUFFER, R2	
			0D	12 0000C	TSTL	72(AP)	0746
		00000000G	00	9F 0000E	BNEQ	1\$	
00000000G	00		01	FB 00014	PUSHAB	SMG\$ MISTERNAM	0748
	62		6C	FA 0001B	CALLS	#1, [IB\$STOP	
	54		01	D0 0001E	CALLG	(AP), SMG\$\$FLUSH_SAVED_BUFFER	0758
	58		01	D0 0001E	MOVL	#1, 84(AP)	0764
		DA	AF	9E 00022	MOVAB	P.AAA, 88(AP)	0765
	62		6C	FA 00027	CALLG	(AP), SMG\$\$FLUSH_SAVED_BUFFER	0767
	50		01	D0 0002A	MOVL	#1, R0	0769
			04	0002D	RET		0771

; Routine Size: 46 bytes, Routine Base: _SMG\$CODE + 003D

```
0772 1 %SBTTL 'EXPRESSION_ERROR - Signal and expression error'
0773 1 ROUTINE EXPRESSION_ERROR =
0774 1
0775 1 ++
0776 1 FUNCTIONAL DESCRIPTION:
0777 1
0778 1     Signal that an invalid arithmetic expression was found.
0779 1
0780 1 CALLING SEQUENCE:
0781 1
0782 1     status = EXPRESSION_ERROR ()
0783 1
0784 1 FORMAL PARAMETERS:
0785 1
0786 1     NONE
0787 1
0788 1 IMPLICIT INPUTS:
0789 1
0790 1     AP     Points to TPARSE parameter block
0791 1
0792 1 IMPLICIT OUTPUTS:
0793 1
0794 1     NONE
0795 1
0796 1 COMPLETION STATUS:
0797 1
0798 1     $$$_NORMAL
0799 1
0800 1 SIDE EFFECTS:
0801 1
0802 1 --
0803 1
0804 1 BEGIN
0805 2 BUILTIN
0806 2
0807 2     AP;
0808 2
0809 2     MAP
0810 2         AP : REF BLOCK [,BYTE];
0811 2
0812 2     SIGNAL_STOP (SMG$ ERRAT LIN,
0813 2                 3, .SMG$$CURRENT LINE,
0814 2                 .AP [TPASL-TOKENCNT],
0815 2                 .AP [TPASL-TOKENPTR],
0816 2                 SMG$_INVEXP)
0817 1
0817 1 END;                                ! end of routine EXPRESSION_ERROR
```

```
0000 00000 EXPRESSION_ERROR:
                                .WORD    Save nothing
00000000G 00 9F 00002          PUSHAB  SMG$ INVEXP
7E          10 AC 7D 00008          MOVQ   16(AP), -(SP)
00000000G 00 DD 0000C          PUSHL   SMG$$CURRENT_LINE
                                03 DD 00012          PUSHL   #3
```

```
0773
0811
0813
0812
0811
```


SMG\$STRING_TABL
1-003

TPARSE tables for string capabilities
EXPRESSION_ERROR - Signal and expression error

D 14
16-Sep-1984 01:22:35
14-Sep-1984 13:10:04

VAX-11 Bliss-32 V4.0-742
[SMGRTL.SRC]SMGSTRTAB.B32;1

Page 71
(6)

00000000G 00 9F 00014
00000000G 00 06 FB 0001A
04 00021

PUSHAB SMG\$ERRAT LIN
CALLS #6, [IB\$STOP
RET

:
:
: 0817

; Routine Size: 34 bytes, Routine Base: _SMG\$CODE + 006B

```

747 0818 1 %SBTTL 'END_ARITH_CAP - End of arithmetic capability data'
748 0819 1 ROUTINE END_ARITH_CAP =
749 0820 1
750 0821 1 ++
751 0822 1 FUNCTIONAL DESCRIPTION:
752 0823 1
753 0824 1     We have found the end of an arithmetic expression (a right
754 0825 1     parenthesis). An arithmetic expression should be terminated
755 0826 1     by the encoded type SMG$K_STORE, indicating that the final
756 0827 1     value has been computed.
757 0828 1
758 0829 1 CALLING SEQUENCE:
759 0830 1
760 0831 1     status = END_ARITH_CAP ()
761 0832 1
762 0833 1 FORMAL PARAMETERS:
763 0834 1
764 0835 1     NONE
765 0836 1
766 0837 1 IMPLICIT INPUTS:
767 0838 1
768 0839 1     AP      Points to TPARSE parameter block
769 0840 1
770 0841 1 IMPLICIT OUTPUTS:
771 0842 1
772 0843 1     NONE
773 0844 1
774 0845 1 COMPLETION STATUS:
775 0846 1
776 0847 1     SSS_NORMAL
777 0848 1
778 0849 1 SIDE EFFECTS:
779 0850 1
780 0851 1 --
781 0852 1
782 0853 1 BEGIN
783 0854 2
784 0855 2 BUILTIN
785 0856 2 CALLG,
786 0857 2 AP;
787 0858 2 MAP
788 0859 2 AP : REF BLOCK [,BYTE];
789 0860 2
790 0861 2 ++
791 0862 2 Indicate end of computing.
792 0863 2 --
793 0864 2
794 0865 2 AP [PARAM_L_SAVED_TOKENCNT] = 1;
795 0866 2 AP [PARAM_L_SAVED_TOKENSTR] = UPLIT (BYTE (SMG$K_STORE));
796 0867 2
797 0868 2 CALLG (.AP, SMG$$FLUSH_ARITHMETIC);
798 0869 2 ! move the data
799 0870 2
800 0871 2 AP [TPASV_BLANKS] = 1;
801 0872 2 ! look at blanks again
802 0873 2
803 0874 2 RETURN (SS$_NORMAL);

```

```

SMG$STRING_TABL TPARSE tables for string capabilities
1-003          END_ARITH_CAP - End of arithmetic capability da
; 804          0875 1      END;

```

```

F 14
16-Sep-1984 01:22:35
14-Sep-1984 13:10:04

```

```

VAX-11 Bliss-32 V4.0-742
[SMGRTL.SRC]SMGSTRTAB.B32;1

```

```

Page 73
(7)

```

```

! end of routine END_ARITH_CAP

```

```

F7 0008D .BLKB 3
00090 P.AAB: .BYTE -9

```

```

0000 00000 END_ARITH_CAP:
54 AC 01 D0 00002 .WORD Save nothing : 0819
58 AC AF 9E 00006 MOVL #1, 84(AP) : 0865
00000000G 00 F6 6C FA 0000B MOVAB P.AAB, 88(AP) : 0866
04 AC 01 88 00012 CALLG (AP), SMG$$FLUSH_ARITHMETIC : 0868
50 01 D0 00016 BISB2 #1, 4(AP) : 0871
04 00019 MOVL #1, R0 : 0873
RET : 0875

```

```

; Routine Size: 26 bytes, Routine Base: _SMG$CODE + 0091

```

```
006 0876 1 XSBTTL 'INIT_ARITH_CAP - Initialize arithmetic capability data'
007 0877 1 ROUTINE INIT_ARITH_CAP =
008 0878 1
009 0879 1 ++
010 0880 1 FUNCTIONAL DESCRIPTION:
011 0881 1
012 0882 1     We have found the beginning of an arithmetic expression (a left
013 0883 1     parenthesis). Before moving operands and operators into the data
014 0884 1     area, initialize it as an arithmetic capability, ie. start with a
015 0885 1     negative length and a type of SMG$K_ARITH_STRING.
016 0886 1
017 0887 1 CALLING SEQUENCE:
018 0888 1
019 0889 1     status = INIT_ARITH_CAP ()
020 0890 1
021 0891 1 FORMAL PARAMETERS:
022 0892 1
023 0893 1     NONE
024 0894 1
025 0895 1 IMPLICIT INPUTS:
026 0896 1
027 0897 1     AP      Points to TPARSE parameter block
028 0898 1
029 0899 1 IMPLICIT OUTPUTS:
030 0900 1
031 0901 1     NONE
032 0902 1
033 0903 1 COMPLETION STATUS:
034 0904 1
035 0905 1     SS$_NORMAL
036 0906 1
037 0907 1 SIDE EFFECTS:
038 0908 1
039 0909 1 --
040 0910 1
041 0911 2 BEGIN
042 0912 2
043 0913 2 BUILTIN
044 0914 2     CALLG,
045 0915 2     AP;
046 0916 2 MAP
047 0917 2     AP : REF BLOCK [,BYTE];
048 0918 2
049 0919 2 LOCAL
050 0920 2     FIRST : INITIAL (0),          ! flag to indicate start of string
051 0921 2     START_CAP_STRING;             ! where to copy data
052 0922 2 MAP
053 0923 2     START_CAP_STRING : REF VECTOR [,BYTE,SIGNED];
054 0924 2
055 0925 2 BIND
056 0926 2     CAP_PTRS = .AP [PARAM_L_CUR_TERM_DEF] : VECTOR [,WORD];
057 0927 2
058 0928 2
059 0929 2 ++
060 0930 2     The NAME capability should have preceeded any arithmetic capability.
061 0931 2     Complain if it didn't.
062 0932 2 --
```



```

863 0933 2 IF CAP_PTRS EQL 0
864 0934 THEN
865 0935 SIGNAL_STOP (SMG$_MISTERNAM);
866 0936
867 0937
868 0938
869 0939
870 0940
871 0941
872 0942
873 0943 CALLG (.AP, SMG$$FLUSH_SAVED_BUFFER);
874 0944
875 0945 IF .CAP_PTRS [.AP [PARAM_L_CUR_CAP_NUMBER]] NEQ 0
876 0946 THEN
877 0947 BEGIN
878 0948
879 0949
880 0950
881 0951
882 0952
883 0953
884 0954
885 0955
886 0956
887 0957
888 0958
889 0959
890 0960
891 0961
892 0962
893 0963
894 0964
895 0965
896 0966
897 0967
898 0968
899 0969
900 0970
901 0971
902 0972
903 0973
904 0974
905 0975
906 0976
907 0977
908 0978
909 0979
910 0980
911 0981
912 0982
913 0983
914 0984
915 0985
916 0986
917 0987
918 0988
919 0989

+ We may have copied some normal text before we encountered an
+ arithmetic expression. Find the start of the string, and make
+ sure that there isn't some more saved text.
-

CALLG (.AP, SMG$$FLUSH_SAVED_BUFFER);

IF .CAP_PTRS [.AP [PARAM_L_CUR_CAP_NUMBER]] NEQ 0
THEN
BEGIN
+ Not start of the capability string.
-
START_CAP_STRING = .AP [PARAM_L_CUR_TERM_DEF] +
.CAP_PTRS [.AP [PARAM_L_CUR_CAP_NUMBER]];
END
ELSE
+ First byte of capability string. Set offset pointer.
-
BEGIN
FIRST = 1;
START_CAP_STRING = .AP [PARAM_L_CUR_DATA_BYTE];
CAP_PTRS [.AP [PARAM_L_CUR_CAP_NUMBER]] = .SMG$$DATA_OFFSET;
SMG$$DATA_OFFSET = .SMG$$DATA_OFFSET + 2;
! leave space for at least size, type
! bytes
END;

IF .FIRST
THEN
AP [PARAM_L_CUR_DATA_BYTE] = .AP [PARAM_L_CUR_DATA_BYTE] + 2;
! space for size & type bytes

+ If some other things preceeded this arithmetic expression, we need
+ to move them over by 2 bytes to make room for a size and type.
-

IF NOT .FIRST
THEN
BEGIN
LOCAL
TYPE;
TYPE = .START_CAP_STRING [1]; ! make this a longword
IF .TYPE NEQ SMG$K_ARITH_STRING
THEN
BEGIN
! first expression
+ We don't allow mixing of FAO strings and arithmetic
+ expressions. We know the type isn't arithmetic, so
+ if the length is negative, there must be an FAO string.
-

```

```

920 0990 4      !-
921 0991 4      IF .START_CAP_STRING [0] LSS 0
922 0992 4      THEN
923 0993 4          SIGNAL_STOP (SMG$ERRAT LIN,
924 0994 4              3, .SMG$SCURRENT LINE,
925 0995 4              .AP [PARAM_L_SAVED_TOKENCNT],
926 0996 4              .AP [PARAM_L_SAVED_TOKENSTR],
927 0997 4              SMG$_SYNERR);
928 0998 4
929 0999 4      !+
930 1000 4      ! Slide over text.
931 1001 4      !-
932 1002 4      CHSMOVE (.START_CAP_STRING [0] + 2, ! # chars + size, type bytes
933 1003 4          START_CAP_STRING [0], ! from old location
934 1004 4          START_CAP_STRING [2]); ! to new
935 1005 4
936 1006 4      !+
937 1007 4      ! Need a negative size. The size includes the data,
938 1008 4      ! the size byte and the type byte.
939 1009 4
940 1010 4      ! Note that right now, the size of the normal text is the size of
941 1011 4      ! the entire capability string.
942 1012 4      !-
943 1013 4      IF .START_CAP_STRING [0] GTR 0
944 1014 4      THEN
945 1015 4          START_CAP_STRING [0] = -.START_CAP_STRING [0] - 4;
946 1016 4              ! size + 2 size & type bytes
947 1017 4
948 1018 4      AP [PARAM_L_CUR_DATA_BYTE] = .AP [PARAM_L_CUR_DATA_BYTE] + 2;
949 1019 4          ! update next free byte
950 1020 4      SMG$$DATA_OFFSET = .SMG$$DATA_OFFSET + 2;
951 1021 4          ! update next offset
952 1022 4      END; ! first expression
953 1023 3
954 1024 3      END ! not first byte in capability
955 1025 2
956 1026 2      ELSE
957 1027 2          START_CAP_STRING [0] = .START_CAP_STRING [0] - 2;
958 1028 2              ! include size & type bytes for
959 1029 2              ! arith in count
960 1030 2
961 1031 2      START_CAP_STRING [1] = SMG$K_ARITH_STRING <0,8>;
962 1032 2          ! set (or reset) as arith string
963 1033 2
964 1034 2      AP [TPASV_BLANKS] = 0; ! ignore blanks between ( )
965 1035 2
966 1036 2      RETURN (SS$_NORMAL);
967 1037 2
968 1038 1      END; ! end of routine INIT_ARITH_CAP
  
```

01FC 00000 INIT_ARITH_CAP:
 58 00000000G 00 9E 00002 .WORD Save R2,R3,R4,R5,R6,R7,R8
 MOVAB LIB\$STOP, R8

: 0877
 :

	57	00000000G	00	9E	00009	MOVAB	SMG\$DATA_OFFSET, R7		
			53	D4	00010	CLRL	FIRST	0911	
	52	48	AC	D0	00012	MOVL	72(AP), R2	0926	
			09	12	00016	BNEQ	1\$	0933	
		00000000G	00	9F	00018	PUSHAB	SMG\$ MISTERNAM	0935	
	68		01	FB	0001E	CALLS	#1, [IB\$STOP		
00000000G	00		6C	FA	00021	CALLG	(AP), SMG\$FLUSH_SAVED_BUFFER	0943	
	50	50	AC	D0	00028	MOVL	80(AP), R0	0945	
			6240	B5	0002C	TSTW	(R2)[R0]		
			0A	13	0002F	BEQL	2\$		
	56		6240	3C	00031	MOVZWL	(R2)[R0], START_CAP_STRING	0952	
	56	48	AC	C0	00035	ADDL2	72(AP), START_CAP_STRING		
			0E	11	00039	BRB	3\$	0945	
	53		01	D0	0003B	MOVL	#1, FIRST	0959	
	56	4C	AC	D0	0003E	MOVL	76(AP), START_CAP_STRING	0960	
6240			67	B0	00042	MOVW	SMG\$DATA_OFFSET, -(R2)[R0]	0961	
	67		02	C0	00046	ADDL2	#2, SMG\$DATA_OFFSET	0962	
	07		53	E9	00049	BLBC	FIRST, 4\$	0967	
4C	AC		02	C0	0004C	ADDL2	#2, 76(AP)	0969	
	4A		53	E8	00050	BLBS	FIRST, 7\$	0981	
FFFFFFFE	50	01	A6	98	00053	CVTBL	1(START_CAP_STRING), TYPE		
	8F		50	D1	00057	CMPL	TYPE, #=2	0983	
			40	13	0005E	BEQL	8\$		
			66	95	00060	TSTB	(START_CAP_STRING)	0991	
			1B	18	00062	BGEQ	5\$		
		00000000G	00	9F	00064	PUSHAB	SMG\$ SYNERR	0993	
	7E	54	AC	7D	0006A	MOVQ	84(AP), -(SP)	0995	
		00000000G	00	DD	0006E	PUSHL	SMG\$CURRENT_LINE	0994	
			03	DD	00074	PUSHL	#3	0993	
		00000000G	00	9F	00076	PUSHAB	SMG\$ ERRAT LIN		
	68		06	FB	0007C	CALLS	#6, [IB\$STOP		
	50		66	98	0007F	CVTBL	(START_CAP_STRING), R0	1002	
	50		02	C0	00082	ADDL2	#2, R0		
02	A6		50	28	00085	MOVCS	R0, (START_CAP_STRING), 2(START_CAP_STRING)	1004	
	66		66	95	0008A	TSTB	(START_CAP_STRING)	1013	
			06	15	0008C	BLEQ	6\$		
	66		66	8E	0008E	MNEGB	(START_CAP_STRING), (START_CAP_STRING)	1015	
	66		04	82	00091	SUBB2	#4, (START_CAP_STRING)		
4C	AC		02	C0	00094	ADDL2	#2, 76(AP)	1018	
	67		02	C0	00098	ADDL2	#2, SMG\$DATA_OFFSET	1020	
			03	11	0009B	BRB	8\$	0976	
	66		02	82	0009D	SUBB2	#2, (START_CAP_STRING)	1026	
01	A6		02	8E	000A0	MNEGB	#2, 1(START_CAP_STRING)	1031	
04	AC		01	8A	000A4	BICB2	#1, 4(AP)	1034	
	50		01	D0	000A8	MOVL	#1, R0	1036	
			04	00	000AB	RET		1038	

; Routine Size: 172 bytes, Routine Base: _SMG\$CODE + 00AB

```

970 1039 1 %SBTTL 'INSERT_CARROT - Insert a ^ in the capability string'
971 1040 1 ROUTINE INSERT_CARROT =
972 1041 1
973 1042 1 ++
974 1043 1 FUNCTIONAL DESCRIPTION:
975 1044 1
976 1045 1     Stores a ^ character in the current capability string in
977 1046 1     TERMTABLE.EXE.
978 1047 1
979 1048 1 CALLING SEQUENCE:
980 1049 1
981 1050 1     status = INSERT_CARROT ()
982 1051 1
983 1052 1 FORMAL PARAMETERS:
984 1053 1
985 1054 1     NONE
986 1055 1
987 1056 1 IMPLICIT INPUTS:
988 1057 1
989 1058 1     AP     Points to TPARSE parameter block
990 1059 1
991 1060 1 IMPLICIT OUTPUTS:
992 1061 1
993 1062 1     NONE
994 1063 1
995 1064 1 COMPLETION STATUS:
996 1065 1
997 1066 1     $$$_NORMAL
998 1067 1
999 1068 1 SIDE EFFECTS:
1000 1069 1
1001 1070 1 --
1002 1071 1
1003 1072 2 BEGIN
1004 1073 2
1005 1074 2 BUILTIN
1006 1075 2 CALLG,
1007 1076 2 AP;
1008 1077 2 MAP
1009 1078 2 AP : REF BLOCK [,BYTE];
1010 1079 2
1011 1080 2 ++
1012 1081 2 If this is not the NAME capability and we have no pointers set up
1013 1082 2 for the terminal definition, then NAME was not the first capability
1014 1083 2 in the definition. Complain.
1015 1084 2 --
1016 1085 2
1017 1086 2 BEGIN
1018 1087 2 BIND
1019 1088 2 CAP_PTRS = .AP [PARAM_L_CUR_TERM_DEF] : VECTOR [,WORD];
1020 1089 2
1021 1090 2 IF CAP_PTRS EQL 0
1022 1091 2 THEN
1023 1092 2 SIGNAL_STOP (SMG$_MISTERNAM);
1024 1093 2
1025 1094 2 ++
1026 1095 2 Move the capability data. The byte count is in the first byte and

```



```

1027 1096 3 | the actual data follows.
1028 1097 3 |
1029 1098 3 | Part of the string may already be copied - append in this part so
1030 1099 3 | as not to overwrite it. (This can happen if escape or control are
1031 1100 3 | part of the sequence.)
1032 1101 3 |
1033 1102 3 |
1034 1103 3 | CALLG (.AP, SMG$$FLUSH_SAVED_BUFFER);
1035 1104 3 | ! copy saved string
1036 1105 3 | AP [PARAM_L_SAVED_TOKENCNT] = 1;
1037 1106 3 | AP [PARAM_L_SAVED_TOKENSTR] = UPLIT ('^');
1038 1107 3 | ! carrot is now 'saved'
1039 1108 3 | CALLG (.AP, SMG$$FLUSH_SAVED_BUFFER);
1040 1109 3 |
1041 1110 3 | END; ! end of BINDs scope
1042 1111 3 |
1043 1112 3 | RETURN (SS$ _NORMAL);
1044 1113 3 |
1045 1114 3 | END; ! end of routine INSERT_CARROT

```

```

00 00 00 5E 00157 .BLKB 1
00158 P.AAC: .ASCII \^\<0><0><0>

```

```

0004 00000 INSERT_CARROT:
52 00000000G 00 9E 00002 .WORD Save R2 1040
48 AC D5 00009 MOVAB SMG$$FLUSH_SAVED_BUFFER, R2
0D 12 0000C TSTL 72(AP) 1090
00000000G 00 9F 0000E BNEQ 1$
00000000G 01 FB 00014 PUSHAB SMG$ _MISTERNAM 1092
54 62 01 FA 0001B 1$: CALLS #1, [IB$STOP
58 AC 01 D0 0001E CALLG (AP), SMG$$FLUSH_SAVED_BUFFER 1103
62 D7 AF 9E 00022 MOVL #1, 84(AP) 1105
50 6C FA 00027 MOVAB P.AAC, 88(AP) 1106
01 D0 0002A CALLG (AP), SMG$$FLUSH_SAVED_BUFFER 1108
04 0002D MOVL #1, R0 1112
RET 1114

```

; Routine Size: 46 bytes, Routine Base: _SMG\$CODE + 015C

```
1047 1115 1 %SBTTL 'INSERT_DOLLAR - Insert a $ into a capability string'
1048 1116 1 ROUTINE INSERT_DOLLAR =
1049 1117 1
1050 1118 1 ++
1051 1119 1 FUNCTIONAL DESCRIPTION:
1052 1120 1
1053 1121 1     Stores a $ character in the current capability string in
1054 1122 1     TERMTABLE.EXE.
1055 1123 1
1056 1124 1 CALLING SEQUENCE:
1057 1125 1
1058 1126 1     status = INSERT_DOLLAR ()
1059 1127 1
1060 1128 1 FORMAL PARAMETERS:
1061 1129 1
1062 1130 1     NONE
1063 1131 1
1064 1132 1 IMPLICIT INPUTS:
1065 1133 1
1066 1134 1     AP      Points to TPARSE parameter block
1067 1135 1
1068 1136 1 IMPLICIT OUTPUTS:
1069 1137 1
1070 1138 1     NONE
1071 1139 1
1072 1140 1 COMPLETION STATUS:
1073 1141 1
1074 1142 1     SS$_NORMAL
1075 1143 1
1076 1144 1 SIDE EFFECTS:
1077 1145 1
1078 1146 1 --
1079 1147 1
1080 1148 2 BEGIN
1081 1149 2
1082 1150 2 BUILTIN
1083 1151 2 CALLG,
1084 1152 2 AP;
1085 1153 2 MAP
1086 1154 2 AP : REF BLOCK [,BYTE];
1087 1155 2 BIND
1088 1156 2 CAP_PTRS = .AP [PARAM_L_CUR_TERM_DEF] : VECTOR [,WORD];
1089 1157 2
1090 1158 2 ++
1091 1159 2 If this is not the NAME capability and we have no pointers set up
1092 1160 2 for the terminal definition, then NAME was not the first capability
1093 1161 2 in the definition. Complain.
1094 1162 2 --
1095 1163 2
1096 1164 2 IF CAP_PTRS EQL 0
1097 1165 2 THEN
1098 1166 2     SIGNAL_STOP (SMG$_MISTERNAM);
1099 1167 2
1100 1168 2 ++
1101 1169 2 Move the capability data. The byte count is in the first byte and
1102 1170 2 the actual data follows.
1103 1171 2
```

```

1104 1172 2 | Part of the string may already be copied - append in this part so
1105 1173 2 | as not to overwrite it. (This can happen if escape or control are
1106 1174 2 | part of the sequence.)
1107 1175 2 |
1108 1176 2 |
1109 1177 2 | CALLG (.AP, SMG$$FLUSH_SAVED_BUFFER);
1110 1178 2 | ! copy saved string
1111 1179 2 | AP [PARAM_L_SAVED_TOKENCNT] = 1;
1112 1180 2 | AP [PARAM_L_SAVED_TOKENSTR] = UPLIT ('$');
1113 1181 2 | ! $ is now 'saved'
1114 1182 2 | CALLG (.AP, SMG$$FLUSH_SAVED_BUFFER);
1115 1183 2 | ! append $
1116 1184 2 |
1117 1185 2 | RETURN (SS$_NORMAL);
1118 1186 2 |
1119 1187 2 | END;

```

! end of routine INSERT_DOLLAR

00 00 00 24 0018A .BLKB 2
 0018C P.AAD: .ASCII \\$\<0><0><0>

				0004 00000	INSERT_DOLLAR:		
	52	00000000G	00	9E 00002	.WORD	Save R2	1116
		48	AC	D5 00009	MOVAB	SMG\$\$FLUSH_SAVED_BUFFER, R2	
			0D	12 0000C	TSTL	72(AP)	1164
		00000000G	00	9F 0000E	BNEQ	1\$	
00000000G	00		01	FB 00014	PUSHAB	SMG\$ MISTERNAM	1166
	62		6C	FA 0001B	CALLS	#1, [IB\$STOP	
	54	AC	01	FA 0001B	CALLG	(AP), SMG\$\$FLUSH_SAVED_BUFFER	1177
	58	AC	01	D0 0001E	MOVL	#1, 84(AP)	1179
		D7	AF	9E 00022	MOVAB	P.AAD, 88(AP)	1180
	62		6C	FA 00027	CALLG	(AP), SMG\$\$FLUSH_SAVED_BUFFER	1182
	50		01	D0 0002A	MOVL	#1, R0	1185
			04	0002D	RET		1187

; Routine Size: 46 bytes, Routine Base: _SMG\$CODE + 0190

```

1121 1188 1 %SBTTL 'INSERT_EXCLAMATION - Insert a ! into a capability string'
1122 1189 1 ROUTINE INSERT_EXCLAMATION =
1123 1190 1
1124 1191 1
1125 1192 1 ++
1126 1193 1 FUNCTIONAL DESCRIPTION:
1127 1194 1     Stores a ! character in the current capability string in
1128 1195 1     TERMTABLE.EXE.
1129 1196 1
1130 1197 1 CALLING SEQUENCE:
1131 1198 1     status = INSERT_EXCLAMATION ()
1132 1199 1
1133 1200 1 FORMAL PARAMETERS:
1134 1201 1
1135 1202 1     NONE
1136 1203 1
1137 1204 1 IMPLICIT INPUTS:
1138 1205 1
1139 1206 1     AP     Points to TPARSE parameter block
1140 1207 1
1141 1208 1 IMPLICIT OUTPUTS:
1142 1209 1
1143 1210 1     NONE
1144 1211 1
1145 1212 1 COMPLETION STATUS:
1146 1213 1
1147 1214 1     SSS_NORMAL
1148 1215 1
1149 1216 1 SIDE EFFECTS:
1150 1217 1
1151 1218 1 --
1152 1219 1
1153 1220 1 BEGIN
1154 1221 2 BUILTIN
1155 1222 2 CALLG,
1156 1223 2 AP;
1157 1224 2
1158 1225 2 MAP
1159 1226 2 AP : REF BLOCK [,BYTE];
1160 1227 2 BIND
1161 1228 2 CAP_PTRS = .AP [PARAM_L_CUR_TERM_DEF] : VECTOR [,WORD];
1162 1229 2
1163 1230 2 ++
1164 1231 2 If this is not the NAME capability and we have no pointers set up
1165 1232 2 for the terminal definition, then NAME was not the first capability
1166 1233 2 in the definition. Complain.
1167 1234 2 --
1168 1235 2
1169 1236 2 IF CAP_PTRS EQL 0
1170 1237 2 THEN
1171 1238 2     SIGNAL_STOP (SMG$_MISTERNAM);
1172 1239 2
1173 1240 2 ++
1174 1241 2 Move the capability data. The byte count is in the first byte and
1175 1242 2 the actual data follows.
1176 1243 2
1177 1244 2 Part of the string may already be copied - append in this part so

```


SMG\$STRING_TABL
1-003

TPARSE tables for string capabilities

INSERT_EXCLAMATION - Insert a ! into a capabili

C 15

16-Sep-1984 01:22:35

14-Sep-1984 13:10:04

VAX-11 Bliss-32 V4.0-742

[SMGRTL.SRC]SMGSTRTAB.B32;1

Page 83

(11)

```

: 1178      1245  2  ! as not to overwrite it. (This can happen if escape or control are
: 1179      1246  2  ! part of the sequence.)
: 1180      1247  2  !
: 1181      1248  2  !
: 1182      1249  2  CALLG (.AP, SMG$$FLUSH_SAVED_BUFFER);
: 1183      1250  2  T copy saved buffer
: 1184      1251  2  AP [PARAM_L_SAVED_TOKENCNT] = 1;
: 1185      1252  2  AP [PARAM_L_SAVED_TOKENSTR] = UPLIT ('!');
: 1186      1253  2  ! exclamation is now 'saved'
: 1187      1254  2  CALLG (.AP, SMG$$FLUSH_SAVED_BUFFER);
: 1188      1255  2  T append !
: 1189      1256  2
: 1190      1257  2  RETURN (SS$_NORMAL);
: 1191      1258  2
: 1192      1259  1  END;
                                ! end of routine INSERT_EXCLAMATION
```

```

                                001BE  .BLKB  2
00 00 00 21 001C0 P.AAE: .ASCII \!\<0><0><0>
```

```

                                0004 00000 INSERT_EXCLAMATION:
                                .WORD  Save R2
52 00000000G 00 9E 00002 MOVAB SMG$$FLUSH_SAVED_BUFFER, R2
                                48 AC D5 00009 TSTL 72(AP)
                                0D 12 0000C BNEQ 1$
00000000G 00 9F 0000E PUSHAB SMG$ MISTERNAM
00 62 01 FB 00014 CALLS #1, [IB$STOP
54 AC 6C FA 0001B 1$: CALLG (AP), SMG$$FLUSH_SAVED_BUFFER
58 AC 01 D0 0001E MOVL #1, 84(AP)
62 D7 AF 9E 00022 MOVAB P.AAE, 88(AP)
50 6C FA 00027 CALLG (AP), SMG$$FLUSH_SAVED_BUFFER
01 D0 0002A MOVL #1, R0
04 0002D RET
```

; Routine Size: 46 bytes, Routine Base: _SMG\$CODE + 01C4

```
1194 1260 1 %SBTTL 'INSERT_PARENTHESIS - Insert a ( into a capability string'
1195 1261 1 ROUTINE INSERT_PARENTHESIS =
1196 1262 1
1197 1263 1 ++
1198 1264 1 FUNCTIONAL DESCRIPTION:
1199 1265 1
1200 1266 1     Stores a ( character in the current capability string in
1201 1267 1     TERMTABLE.EXE.
1202 1268 1
1203 1269 1 CALLING SEQUENCE:
1204 1270 1
1205 1271 1     status = INSERT_PARENTHESIS ()
1206 1272 1
1207 1273 1 FORMAL PARAMETERS:
1208 1274 1
1209 1275 1     NONE
1210 1276 1
1211 1277 1 IMPLICIT INPUTS:
1212 1278 1
1213 1279 1     AP      Points to TPARSE parameter block
1214 1280 1
1215 1281 1 IMPLICIT OUTPUTS:
1216 1282 1
1217 1283 1     NONE
1218 1284 1
1219 1285 1 COMPLETION STATUS:
1220 1286 1
1221 1287 1     SSS_NORMAL
1222 1288 1
1223 1289 1 SIDE EFFECTS:
1224 1290 1
1225 1291 1 --
1226 1292 1
1227 1293 2 BEGIN
1228 1294 2 BUILTIN
1229 1295 2 CALLG,
1230 1296 2 AP;
1231 1297 2 MAP
1232 1298 2 AP : REF BLOCK [,BYTE];
1233 1299 2 BIND
1234 1300 2 CAP_PTRS = .AP [PARAM_L_CUR_TERM_DEF] : VECTOR [,WORD];
1235 1301 2
1236 1302 2 ++
1237 1303 2 If this is not the NAME capability and we have no pointers set up
1238 1304 2 for the terminal definition, then NAME was not the first capability
1239 1305 2 in the definition. Complain.
1240 1306 2 --
1241 1307 2
1242 1308 2 IF CAP_PTRS EQL 0
1243 1309 2 THEN
1244 1310 2     SIGNAL_STOP (SMG$_MISTERNAM);
1245 1311 2
1246 1312 2 ++
1247 1313 2 Move the capability data. The byte count is in the first byte and
1248 1314 2 the actual data follows.
1249 1315 2
1250 1316 2 Part of the string may already be copied - append in this part so
```

```

: 1251      1317 2 | as not to overwrite it. (This can happen if escape or control are
: 1252      1318 2 | part of the sequence.)
: 1253      1319 2 |
: 1254      1320 2 |
: 1255      1321 2 | CALLG (.AP, SMG$$FLUSH_SAVED_BUFFER);
: 1256      1322 2 | T copy saved string
: 1257      1323 2 | AP [PARAM_L_SAVED_TOKENCNT] = 1;
: 1258      1324 2 | AP [PARAM_L_SAVED_TOKENSTR] = UPLIT ('(');
: 1259      1325 2 | ! ( is now saved
: 1260      1326 2 | CALLG (.AP, SMG$$FLUSH_SAVED_BUFFER);
: 1261      1327 2 | T append (
: 1262      1328 2 |
: 1263      1329 2 | RETURN (SS$_NORMAL);
: 1264      1330 2 |
: 1265      1331 1 | END;
                                ! end of routine INSERT_PARENTHESIS

```

```

00 00 00 28 001F2 .BLKB 2
001F4 P.AAF: .ASCII \(\<0><0><0>

```

```

0004 00000 INSERT_PARENTHESIS:
52 00000000G 00 9E 00002 .WORD Save R2
48 AC D5 00009 MOVAB SMG$$FLUSH_SAVED_BUFFER, R2
0D 12 0000C TSTL 72(AP)
00 9F 0000E BNEQ 1$
00000000G 00 01 FB 00014 PUSHAB SMG$ MISTERNAM
62 6C FA 0001B 1$: CALLS #1, [1B$STOP
54 AC 01 D0 0001E CALLG (AP), SMG$$FLUSH_SAVED_BUFFER
58 AC D7 AF 9E 00022 MOVL #1, 84(AP)
62 6C FA 00027 MOVAB P.AAF, 88(AP)
50 01 D0 0002A CALLG (AP), SMG$$FLUSH_SAVED_BUFFER
04 0002D MOVL #1, R0
RET

```

; Routine Size: 46 bytes, Routine Base: _SMG\$CODE + 01F8

```

1267 1332 1 XSBTTL 'INVALID_DIRECTIVE - signal invalid directive error'
1268 1333 1 ROUTINE INVALID_DIRECTIVE =
1269 1334 1
1270 1335 1 ++
1271 1336 1 FUNCTIONAL DESCRIPTION:
1272 1337 1
1273 1338 1     Following a !, we have found some directive we don't handle.
1274 1339 1     Signal an error.
1275 1340 1
1276 1341 1 CALLING SEQUENCE:
1277 1342 1
1278 1343 1
1279 1344 1     status = INVALID_DIRECTIVE ()
1280 1345 1
1281 1346 1 FORMAL PARAMETERS:
1282 1347 1
1283 1348 1     NONE
1284 1349 1
1285 1350 1 IMPLICIT INPUTS:
1286 1351 1
1287 1352 1     AP     Points to TPARSE parameter block
1288 1353 1
1289 1354 1 IMPLICIT OUTPUTS:
1290 1355 1
1291 1356 1     NONE
1292 1357 1
1293 1358 1 COMPLETION STATUS:
1294 1359 1
1295 1360 1     SSS_NORMAL
1296 1361 1
1297 1362 1 SIDE EFFECTS:
1298 1363 1
1299 1364 1 --
1300 1365 1
1301 1366 2 BEGIN
1302 1367 2 BUILTIN
1303 1368 2     AP;
1304 1369 2 MAP
1305 1370 2     AP : REF BLOCK [,BYTE];
1306 1371 2
1307 1372 2 SIGNAL_STOP (SMG$_ERRAT_LIN,
1308 1373 2     3, .SMG$$CURRENT_LINE,
1309 1374 2     .AP [TPASL_TOKENCNT],
1310 1375 2     .AP [TPASL_TOKENPTR],
1311 1376 2     SMG$_INVDIR)
1312 1377 1 END;                                ! end of routine INVALID_DIRECTIVE
  
```

```

                                0000 00000 INVALID_DIRECTIVE:
                                .WORD    Save nothing                : 1333
                                PUSHAB    SMG$_INVDIR                 : 1372
7E 00000000G 00 9F 00002      MOVQ     16(AP), -(SP)                 : 1374
                                00000000G 00 DD 0000C      PUSHL     SMG$$CURRENT_LINE : 1373
                                03 DD 00012      PUSHL     #3           : 1372
  
```


SMG\$STRING_TABL	TPARSE tables for string capabilities	G 15	16-Sep-1984 01:22:35	VAX-11 Bliss-32 V4.0-742	Page 87
1-003	INVALID_DIRECTIVE - signal invalid directive er	14-Sep-1984 13:10:04	[SMGRTL.SRC]SMGSTRTAB.B32;1	(13)	

00000000G	00	9F 00014	PUSHAB	SMG\$ ERRAT LIN	:
	06	FB 0001A	CALLS	#6, CIB\$STOP	:
		04 00021	RET		: 1377

; Routine Size: 34 bytes, Routine Base: _SMG\$CODE + 0226

```

1314 1378 1 XSBTTL 'STORE_OPERAND - Store arithmetic operand'
1315 1379 1 ROUTINE STORE_OPERAND =
1316 1380 1
1317 1381 1 ++
1318 1382 1 FUNCTIONAL DESCRIPTION:
1319 1383 1
1320 1384 1 We have just found an operand which we need to store in TERMTABLE.
1321 1385 1 We will also check to see if an operator was previously found and
1322 1386 1 needs to be stored.
1323 1387 1
1324 1388 1 CALLING SEQUENCE:
1325 1389 1
1326 1390 1 status = STORE_OPERAND ()
1327 1391 1
1328 1392 1 FORMAL PARAMETERS:
1329 1393 1
1330 1394 1 NONE
1331 1395 1
1332 1396 1 IMPLICIT INPUTS:
1333 1397 1
1334 1398 1 AP Points to TPARSE parameter block
1335 1399 1
1336 1400 1 IMPLICIT OUTPUTS:
1337 1401 1
1338 1402 1 NONE
1339 1403 1
1340 1404 1 COMPLETION STATUS:
1341 1405 1
1342 1406 1 SS$NORMAL
1343 1407 1
1344 1408 1 SIDE EFFECTS:
1345 1409 1
1346 1410 1 --
1347 1411 1
1348 1412 1 BEGIN
1349 1413 2 BUILTIN
1350 1414 2 CALLG,
1351 1415 2 AP;
1352 1416 2
1353 1417 2 MAP
1354 1418 2 AP : REF BLOCK [,BYTE];
1355 1419 2 LOCAL
1356 1420 2 COPY_BUFFER : VECTOR [7,BYTE]; ! construct bytes to copy here
1357 1421 2
1358 1422 2 AP [PARAM_L_SAVED_TOKENCNT] = 6; ! longword + type,size bytes
1359 1423 2
1360 1424 2 COPY_BUFFER [0] = SMG$K_OPERAND<0,8>; ! type
1361 1425 2 COPY_BUFFER [1] = 0;
1362 1426 2 CH$MOVE (4, AP [TPA$L_NUMBER], COPY_BUFFER [2]);
1363 1427 2 ! store binary operand
1364 1428 2
1365 1429 2 ++
1366 1430 2 If this is the second operand, then we need to store an operator.
1367 1431 2
1368 1432 2
1369 1433 2 IF .SMG$SMASK_ADR NEQ 0
1370 1434 2 THEN

```

```

1371 1435 3 BEGIN
1372 1436 COPY_BUFFER [6] = .SMG$MASK_ADR;
1373 1437 AP [PARAM_L_SAVED_TOKENCNT] = .AP [PARAM_L_SAVED_TOKENCNT] + 1;
1374 1438 END;
1375 1439
1376 1440 AP [PARAM_L_SAVED_TOKENSTR] = COPY_BUFFER;
1377 1441
1378 1442 CALLG (.AP, SMG$FLUSH_ARITHMETIC); ! write the buffer
1379 1443
1380 1444 SMG$MASK_ADR = 0; ! re-init operator
1381 1445
1382 1446 RETURN (SS$NORMAL);
1383 1447
1384 1448 END; ! end of routine STORE_OPERAND

```

```

0004 0000 STORE_OPERAND:
52 00000000G 00 9E 00002 .WORD Save R2
5E 08 C2 00009 MOVAB SMG$MASK_ADR, R2
54 AC 06 D0 0000C SUBL2 #8, SP
6E FD 8F 9B 00010 MOVL #6, 84(AP)
02 AE 1C AC D0 00014 MOVZBW #253, COPY_BUFFER
50 62 D0 00019 MOVL 28(AP), COPY_BUFFER+2
06 AE 50 90 0001E MOVL SMG$MASK_ADR, R0
58 AC 54 AC D6 00022 BEQL 1$
00000000G 00 6E 9E 00025 1$: MOVAB COPY_BUFFER, 88(AP)
6C FA 00029 CALLG (AP), SMG$FLUSH_ARITHMETIC
62 D4 00030 CLRL SMG$MASK_ADR
50 01 D0 00032 MOVL #1, R0
04 00035 RET

```

; Routine Size: 54 bytes, Routine Base: _SMG\$CODE + 0248

```
1386 1449 1 XSBTTL 'STORE_SUBSTITUTION - store substitution flag in arithmetic cap'
1387 1450 1 ROUTINE STORE_SUBSTITUTION =
1388 1451 1
1389 1452 1
1390 1453 1
1391 1454 1
1392 1455 1
1393 1456 1
1394 1457 1
1395 1458 1
1396 1459 1
1397 1460 1
1398 1461 1
1399 1462 1
1400 1463 1
1401 1464 1
1402 1465 1
1403 1466 1
1404 1467 1
1405 1468 1
1406 1469 1
1407 1470 1
1408 1471 1
1409 1472 1
1410 1473 1
1411 1474 1
1412 1475 1
1413 1476 1
1414 1477 1
1415 1478 1
1416 1479 1
1417 1480 1
1418 1481 1
1419 1482 1
1420 1483 2
1421 1484 2
1422 1485 2
1423 1486 2
1424 1487 2
1425 1488 2
1426 1489 2
1427 1490 2
1428 1491 2
1429 1492 2
1430 1493 2
1431 1494 2
1432 1495 2
1433 1496 2
1434 1497 2
1435 1498 2
1436 1499 2
1437 1500 2
1438 1501 2
1439 1502 2
1440 1503 3
1441 1504 3
1442 1505 3

++
FUNCTIONAL DESCRIPTION:
    Store an encoded flag indicating that a user argument must be
    substituted into the expression at run-time.

CALLING SEQUENCE:
    status = STORE_SUBSTITUTION ()

FORMAL PARAMETERS:
    NONE

IMPLICIT INPUTS:
    AP      Points to TPARSE parameter block

IMPLICIT OUTPUTS:
    NONE

COMPLETION STATUS:
    SSS_NORMAL

SIDE EFFECTS:
--
BEGIN
BUILTIN
CALLG,
AP;
MAP
    AP : REF BLOCK [,BYTE];
LOCAL
    COPY_BUFFER : VECTOR [3,BYTE];

    COPY_BUFFER [0] = SMG$K SUBSTITUTE <0,8>;
    COPY_BUFFER [1] = .AP [TPASL_NUMBER];
    ! store decimal argument number as a byte
    AP [PARAM_L_SAVED_TOKENCNT] = 2;

++
See if there's a saved operator that needs to be inserted.
--

IF .SMG$MASK_ADR NEQ 0
THEN
    BEGIN
        COPY_BUFFER [2] = .SMG$MASK_ADR;
        AP [PARAM_L_SAVED_TOKENCNT] = .AP [PARAM_L_SAVED_TOKENCNT] + 1;
```


1443	1506	3	SMG\$MASK_ADR = 0;
1444	1507	2	END;
1445	1508	2	
1446	1509	2	AP [PARAM_L_SAVED_TOKENSTR] = COPY_BUFFER;
1447	1510	2	
1448	1511	2	CALLG (.AP, SMG\$\$FLUSH_ARITHMETIC);
1449	1512	2	! move buffer to data area
1450	1513	2	
1451	1514	2	RETURN (SS\$_NORMAL);
1452	1515	2	
1453	1516	1	END; ! end of routine STORE_SUBSTITUTION

				0004 00000 STORE_SUBSTITUTION:		
	52	00000000G	00	9E 00002	.WORD Save R2	1450
	5E		04	C2 00009	MOVAB SMG\$MASK_ADR, R2	
	6E		04	8E 0000C	SUBL2 #4, SP	1492
01	AE	1C	AC	90 0000F	MNEGB #4, COPY_BUFFER	1493
54	AC		02	D0 00014	MOVB 28(AP), COPY_BUFFER+1	1495
	50		62	D0 00018	MOVL #2, 84(AP)	1501
			09	13 0001B	MOVL SMG\$MASK_ADR, R0	
02	AE		50	90 0001D	BEQL 1\$	1504
		54	AC	D6 00021	MOVB R0, COPY_BUFFER+2	1505
			62	D4 00024	INCL 84(AP)	1506
58	AC		6E	9E 00026	CLRL SMG\$MASK_ADR	1509
00000000G	00		6C	FA 0002A	MOVAB COPY_BUFFER, 88(AP)	1511
	50		01	D0 00031	CALLG (AP), SMG\$\$FLUSH_ARITHMETIC	1514
			04	00034	MOVL #1, R0	1516
					RET	

; Routine Size: 53 bytes, Routine Base: _SMG\$CODE + 027E

```

: 1455      1517 1 %SBTTL 'NOT_STRING - signal an unknown capability name'
: 1456      1518 1 ROUTINE NOT_STRING =
: 1457      1519 1
: 1458      1520 1 ++
: 1459      1521 1 FUNCTIONAL DESCRIPTION:
: 1460      1522 1
: 1461      1523 1     We just found a unknown capability name. It could be a misspelling
: 1462      1524 1     or it could be a name misplaced under the wrong heading. Signal an
: 1463      1525 1     error.
: 1464      1526 1
: 1465      1527 1 CALLING SEQUENCE:
: 1466      1528 1
: 1467      1529 1     status = NOT_STRING ()
: 1468      1530 1
: 1469      1531 1 FORMAL PARAMETERS:
: 1470      1532 1
: 1471      1533 1     NONE
: 1472      1534 1
: 1473      1535 1 IMPLICIT INPUTS:
: 1474      1536 1
: 1475      1537 1     AP     Points to TPARSE parameter block
: 1476      1538 1
: 1477      1539 1 IMPLICIT OUTPUTS:
: 1478      1540 1
: 1479      1541 1     NONE
: 1480      1542 1
: 1481      1543 1 COMPLETION STATUS:
: 1482      1544 1
: 1483      1545 1     $$$_NORMAL
: 1484      1546 1
: 1485      1547 1 SIDE EFFECTS:
: 1486      1548 1
: 1487      1549 1 --
: 1488      1550 1
: 1489      1551 1 BEGIN
: 1490      1552 2 BUILTIN
: 1491      1553 2 AP:
: 1492      1554 2 MAP
: 1493      1555 2 AP : REF BLOCK [,BYTE];
: 1494      1556 2
: 1495      1557 2 SIGNAL_STOP (SMG$_ERRAT LIN,
: 1496      1558 2 3, .SMG$$CURRENT LINE,
: 1497      1559 2 .AP [TPASL_TOKENCNT],
: 1498      1560 2 .AP [TPASL_TOKENPTR],
: 1499      1561 2 SMG$_NOTSTRCAP)
: 1500      1562 2
: 1501      1563 1 END;                                ! end of routine NOT_STRING

```

		0000 00000	NOT_STRING:				
		00000000G	00 9F 00002	.WORD	Save nothing	:	1518
		7E 00000000G	00 AC 7D 00008	PUSHAB	SMG\$_NOTSTRCAP	:	1558
				MOVQ	16(AP), -(SP)	:	1560
				PUSHL	SMG\$\$CURRENT_LINE	:	1559

SMG\$STRING_TABL	TPARSE tables for string capabilities	M 15				
1-003	NOT_STRING - signal an unknown capability name	16-Sep-1984 01:22:35	VAX-11 Bliss-32 V4.0-742			Page 93
		14-Sep-1984 13:10:04	[SMGRTL.SRC]SMGSTRTAB.B32;1			(16)
		03 DD 00012	PUSHL #3			: 1558
	00000000G 00 00000000G	00 9F 00014	PUSHAB SMG\$ERRAT LIN			: 1563
		06 FB 0001A	CALLS #6, [IB\$STOP			
		04 00021	RET			
; Routine Size: 34 bytes, Routine Base: _SMG\$CODE + 02B3						
; 1502 1564 1 !<BLF/PAGE>						

SMG\$STRING_TABL TPARSE tables for string capabilities
1-003 NOT_STRING - signal an unknown capability name

N 15

16-Sep-1984 01:22:35

VAX-11 Bliss-32 V4.0-742
[SMGRTL.SRC]SMGSTRTAB.B32;1

Page 94
(17)

: 1504 1565 1 END
: 1505 1566 1
: 1506 1567 0 ELUDOM

! End of module SMG\$STRING_TABLES

.EXTRN LIB\$STOP

PSECT SUMMARY

Name	Bytes	Attributes
_LIB\$KEY0\$	342	NOVEC,NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC,ALIGN(1)
_LIB\$STATES	2576	NOVEC,NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC,ALIGN(1)
_LIB\$KEY1\$	2061	NOVEC,NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC,ALIGN(1)
_SMG\$CODE	725	NOVEC,NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	172	1	581	00:01.0
_\$255\$DUA28:[SMGRTL.OBJ]RTLLIB.L32;1	36	0	0	8	00:00.1
_\$255\$DUA28:[SMGRTL.OBJ]SMGLIB.L32;1	469	0	0	38	00:00.4
_\$255\$DUA28:[SMGRTL.OBJ]SMGTPALIB.L32;1	41	14	34	10	00:00.1
_\$255\$DUA28:[SYSLIB]TPAMAC.L32;1	42	30	71	14	00:00.1

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS\$:SMGSTRTAB/OBJ=OBJ\$:SMGSTRTAB MSRC\$:SMGSTRTAB/UPDATE=(ENH\$:SMGSTRTAB
:)

: Size: 694 code + 5010 data bytes
: Run Time: 03:48.4
: Elapsed Time: 10:51.1
: Lines/CPU Min: 411
: Lexemes/CPU-Min:113340
: Memory Used: 758 pages
: Compilation Complete

0361

AH-BT13A-SE
 VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY